May 2004

SERVO MAGAZINE

Let Your Minion Do the Walking

ROBOSAPIEN RISING

Cheap Motion Control

Tetsujin Tech: Pneumatics

05.2004

Vol. 2 No. 5

SERVO

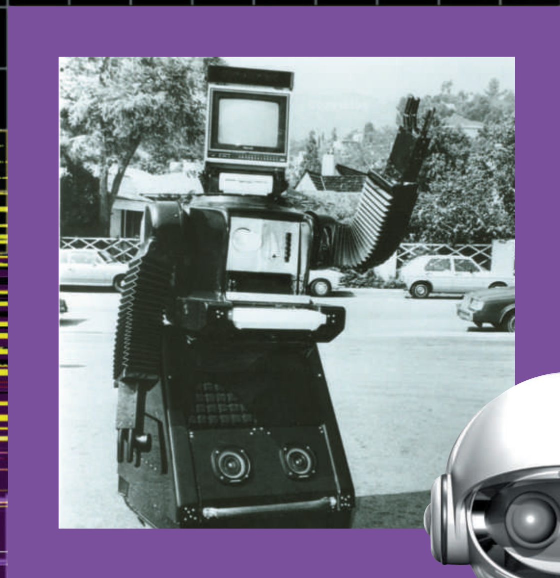
table of contents

In our next issue

Meet Robert Drap and the Odex I

Coming 06.2004

SERVO Magazine (ISSN 1546-0592/CDN Pub Agree#40702530) is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. APPLICATION TO MAIL AT PERIODICALS POSTAGE RATE IS PENDING AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, 430 Princeland Court, Corona, CA 92879-1300** or Station A, P.O. Box 54, Windsor ON N9A 6J5.
cpcreturns@servomagazine.com



features

columns

departments

Mind / Iron

by Dan Danknick

In my March column, I pointed out that engineers frequently cannot give clear explanations for the importance of what they do. I never imagined how substantiated my opinion would be until I read the pop media accounts of the DARPA Grand Challenge — possibly the most technically complex and forward thinking competition ever opened to the public. Here are some of the newspaper headlines that I noticed:

- "Nobody won. Nobody even came close." — Marsha Walton, CNN
- "If They Only Had a Brain" — Mike Langberg, *San Jose Mercury News*
- "Robot Race Suffers Quick, Ignoble End" — Tom Abate, *San Francisco Chronicle*
- "Foiled: DARPA Bots All Fall Down" — The Associated Press

I guess it would be easy to blame these reporters for exhibiting short-sightedness, indulging in sensationalism, or simply wanting to catch an eye with a provocative headline. Very likely, though, the truth of the matter is that the deep difficulty of the DGC was never adequately explained to them, either by the DARPA PR people or by the competitors they interviewed.

Let me give an example that I know all too well from working on The Learning Channel's TV show, *Robotica*:

Say three teammates are slaving over their mechanical creation — pulling wires, wrenching on bolts, and pounding on a laptop. The interviewer asks simply, "What are you guys doing?" The laptop guy replies, "Well, the flashdisk just suffered a CRC error, so we're waiting for the termserver window to open on reboot, then we'll upload a new module via the bootloader." No, this does not need to be translated from the Aramaic, but the interviewer now needs a Tylenol®.

In March, I spent my sixth year as a regional judge for the FIRST competition (**www.usfirst.org**) — where high school students build task-specific robots in a short timeframe. The goal is to get them to interact with local engineers and, hopefully, inspire the study of the hard sciences. I was interviewing a team for a machine quality award and asked simply, "Why did you put round holes in the arm of your robot?" "We have a CNC machine for circular interpolation ..." started off one answer. So I asked the question again. "The arm uses feedback to move to the position that the operator commands ..." This time, the team mentor interjected with, "The judge just wants to know why they are there and why they are round." The light went on. "Oh. They make it lighter without sacrificing strength."

If we want to make an impact on the world and have average people support our efforts, we really need to know when to be in "engineer mode" vs. "normal person mode." I know this may sound a bit odd, but it's true. We'll get much more support from companies that want us to use their products if we can tie the hard core engineering of Young's modulus, interrupt latencies, and tri-state logic to the protection of marine life, relief from repetitive tasks, and helping Grandma walk without a cane.

Of course, like-minded people are often reporters, too. Hans K. Meyer wrote in his March 11, 2004 opinion column for *The Desert Dispatch* that, "this contest ... pits the best minds of corporate and academic America against one another." He concluded that the DGC was a tremendous success, both technically and economically.

With a little thought and a careful answer, articles like his could become the norm and not the exception. SV

Published Monthly By
The TechTrax Group — A Division Of
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(909) 371-8497
FAX **(909) 371-3052**
www.servomagazine.com

Subscription Order ONLY Line
1-800-783-4624

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
robin@servomagazine.com

ADVERTISING SALES DIRECTOR
Rich Collins
rich@servomagazine.com

MANAGING/TECHNICAL EDITOR
Dan Danknick
dan@servomagazine.com

ASSOCIATE EDITOR
Alexandra Lindstrom
alexa@servomagazine.com

CIRCULATION DIRECTOR
Mary Descaro
subscribe@servomagazine.com

WEB CONTENT/STORE
Michael Kaudze
michael@servomagazine.com

PRODUCTION/GRAPHICS
Shannon Lemieux
Glen Pinckney

DATA ENTRY
Janessa Emond
Kristan Rutz

OUR PET ROBOTS
Guido
Mifune

BIO-FEEDBACK

Dear *SERVO*:

I really resonated with Dan Danknick's "Mind/Iron" editorial in the January issue. Yeah, hobbies are where most of us start our journeys to becoming professionals and inventors. I know it was for me and for many of those I've worked with over the years. When I started out, there were far fewer sources of information and materials than there are now. With *SERVO Magazine,* budding roboticists have at their fingertips a valuable source of information, inspiration, and raw materials. Keep up the good work.

Tom McLaughlin
Hicksville, NY

Dear *SERVO*:

I'm puzzled with the changes in direction or did I miss the point? *SERVO* was going to be a magazine on robotics, looking for a way to both show how and tell about robotics. Are there projects in the works? How about starting one over many months, beginning with a basic platform and building up? As people build the project, they could submit their additions or changes. These changes would still have to be compatible or the following months would not work on the main project.

I know vendors would help by wanting to sell parts kits. Start simple and work up, for example:

- Base, motors, power
- Controls, simple sensors
- More advanced sensors and controls
- Radio control from a base
- BASIC Stamp or other controls and programs
- Arms or gripper
- Remote TV link

I like the old robot section, but also want simple project sections.

Thanks for letting me rag and keep up the hard work. I plan on reading every word.

George Jones
Hopewell, VA

The Cutting Edge Robotics project — now on its fifth installment — does a pretty good job of matching your incremental build request.

Alternately, Dafydd Walters' "Open Automaton Project" — featured in the 02.2004 Menagerie — would be a good, though larger, candidate. Perhaps some of the OAP builders could submit some technical reports to us here at SERVO? — Editor Dan

"These *are* the droids you're looking for."

An unidentified member of event security at the 2004 ROBOlympics flaunts his copy of *SERVO Magazine* to Editor Dan. A blaster fight did not ensue, though Dan did keep his thermal grenade close at hand.

Dear *SERVO*:

How about more construction projects on building robots? *SERVO Magazine* has more robotics news than construction projects. Gentlemen, just the first 39 pages deal with *news* articles.

When I want the latest news, I will watch TECHTV.

Louis Fattore
via Internet

Dear *SERVO*:

On one of the Yahoo! groups, a list of robotics clubs is floating around. It sure would be nice if — on your website — you assembled and maintained a list of all the robotics clubs. Make it so there is a hot link to the list on the opening page and even non-subscribers can access it.

Hopefully, sooner or later, builders will come around, really get into robotics and subscribe to *SERVO Magazine.*

Michael Bronosky
via Internet

Done, thanks to Roger Arrick at Arrick Robotics! — Editor Dan

Correction

The April issue's article on LEGO sensors was missing the URL where the sample code could be found. That URL is **www.barello.net/ARC/LEGO**

In addition, the URL for the ARC support group was incorrect and should be **http://groups.yahoo.com/group/AVRRobotControl**

Cheers, Larry Barello

got bot?

Whether you have a build, code, or theory to share, *SERVO* wants to know what you — the resident of the robot workshop — are creating. We want you to Email us your article submissions. Some topics of interest are:

- Sensors and signal processing
- Unique drive geometries
- Mechanical fabrication
- Material selection and use
- Software techniques
- Distributed communication
- Data protocols

Own a Robotic Mi

There´s More to It

by Nicholas Blye

Since the release of i-Cybie, the robot dog, the under $200.00 robotics market hasn't seen much happening in the past couple years — except for Roomba, which is whisking away and vacuuming the nether regions under your couch.

That has changed with the release of RoboSapien.

Developed by Mark W. Tilden and WowWee toys, RoboSapien is the first true consumer biped I've seen. By consumer, I mean it doesn't cost as much as a luxury car (like the Sony Qiro), doesn't come in a kit, is actually available, and can be purchased in a retail store or online.

Wanting a RoboSapien starts as soon as you see it in the box, which, thankfully, doesn't require a screwdriver and an instruction manual to open. (My apologies to any parents who purchased a Tekno for the flashback — just go to your happy place and read on.) Removing some tape and untwisting the retaining wires is all that's needed to get this robot out of its box.

Six Weeks to a Great Body

The design reminds me of a wimpy robot that was strapped to a BowFlex for six weeks. It's a pretty hefty robot, coming in at about four pounds — including the four D batteries — and is very well built.

I can tell you exactly how well-built it is because I disassembled my RoboSapien down to the component level to provide detailed, internal pictures of the innovative, efficient design. I then completely reassembled the robot — without needing a manual or engineering diagram. Excellent internal labeling of the components, connections that are socketed instead of soldered, and a resourceful, intuitive layout made the reassembly much less complicated than it could have been.

Once I put everything back together, it all worked — except for one shoulder that I had reassembled incorrectly. A few screws, a small realignment, and I had gone from a working, factory-fresh RoboSapien to about 300 loose parts lying in a pile, back to a fully working unit again. I haven't seen signs of a switch labeled "Good/Evil: Kill All Humans" anywhere, so I guess we're all fine. (Although that could be in an undocumented IR code!)

Before I tore the poor thing into tiny parts, I tried out a few of the features. The first thing I noticed was the grippers — two different kinds, in fact. Just like Data on *Star Trek*, they are "fully functional."

The robot's right gripper is designed for grasping small objects, such as its included container or even an empty paper towel tube, which it can use to whack something. The left gripper is made just for picking up flat objects, like papers, business cards, or socks.

One of the great things the robot offers is the wide range of built-in swatting, striking, and thumping actions. The best of these actions are under the orange section of commands on the remote. A great deal of thought has gone into this robot and the main method of controlling it is the included remote. Every function is available right out of the box — even programming. We'll look further into operating the remote in a bit.

Att-It-Tude

This is definitely a robot with built-in attitude — if the karate chops and wild roars didn't give you a clue. You'll notice this the first time it wakes up after being sent to sleep.

nion for Just $99.00

Than Meets the Eye

The body itself is just under 14 inches from the feet to the top of the IR-detecting head. You can extend each arm to reach another four and a half inches above the head — that's about 18 inches from the floor. Functional gripper operation will work from about half an inch to 15 inches above the floor.

Technically, there are seven axis motors, but six of them are designed to also operate on an additional axis of movement — providing a total of 13 degrees of freedom. Because of the dual-purpose design, the additional six degrees of freedom aren't completely free, but still give you programming and motion capabilities that would otherwise be impossible in a robot this inexpensive.

If you ever opened or hacked a Furby, you'll recall how many functions were run off a single motor and cam; some of RoboSapien's extended degrees of freedom are based on this same design principle. The grippers, for instance, actuate at the extreme limit of rotation for the arm by cam. While this limits the positions you can use the grippers in, it's important to remember that you actually have functional grippers in a $99.00 robot.

One More Remote

The 21-button remote control operates the grippers, arms, and motions, in addition to doing all programming. It is really an easy-to-use design. All commands printed in red on buttons can be activated at any time by pressing that button. Pressing any of the direction keys, for instance, starts RoboSapien walking or turning in those directions; to run, just press the button a second time.

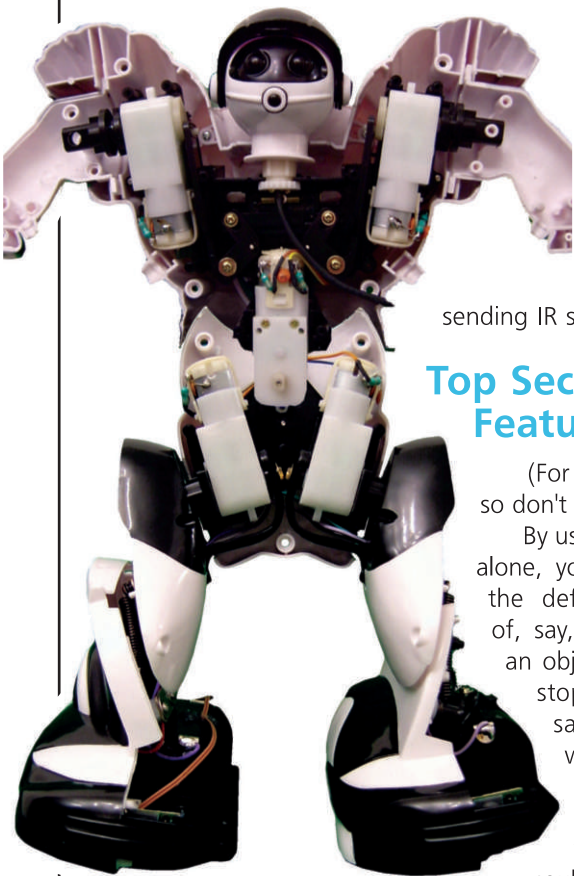
Commands can be interrupted mid-stream, so walking can be switched to a left or right turn or reverse. While moving, you can also execute other functions — like swinging the arms or opening the grippers with each arm moving independently. I recommend that you press Stop if you notice that your terrified cat is all the way down to life number one.

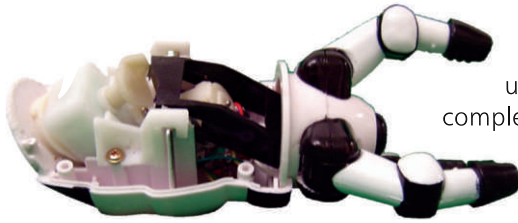
Things get much more interesting when you use the Select button to start exploring the green and orange commands. This is where you begin to really appreciate the command set and simplified remote operation. Considering you're in control of 67 different functions, the interface to those functions is simple and can be grasped in a few seconds.

Press Select once and the light at the top turns green and all of the commands labeled in green can be activated by pressing any key. Press Select twice to see the same idea in action for everything in orange — these are the more aggressive commands, like striking karate chops, sweeping arm motions, and bulldozing forward to shove obstacles out of the way.

RoboSapien includes some built-in programming features — up to 84 steps in three different sensor reaction areas. These areas can be combined and linked to others using the main program. This built-in programming is designed for use with the remote, but it won't be enough to satisfy anyone looking for advanced robot control.

For that, you'll need a different solution and, since RoboSapien uses IR commands, there are a few ways to extend control. One that is being worked on right now involves software running on a PDA that can send IR signals directly to the robot. You can do this yourself with a PDA or even a





cell phone strapped to its back, sending IR signals.

Top Secret Features

(For my eyes only, so don't read this.)

By using the remote alone, you can change the default reactions of, say, bumping into an object. Instead of stopping and saying "Ouch" when it hits something on the left side, your program could be to back up a step, turn away from the object, start walking, and then promptly fart. (As the manual says, advanced technology doesn't mean good manners!)

With your master control, you can link these programs together and let RoboSapien explore, dance, and do just about anything you'd like to try within the built-in actions. One of the best undocumented features I can share with you is an autonomous walking mode that you can program in under one minute.

Grab your remote and press the button to program what happens when a sensor on the right is touched:

```
Press R>
SELECT + SELECT + RIGHT TURN
SELECT + LEAN FORWARD (LEAN FWD)
SELECT + BACK STEP
SELECT + BACK STEP
SELECT + LEFT TURN
SELECT + LEAN BACKWARD (LEAN BWD)
```

Okay, that's a quick 30 seconds, now press the button for the left sensor reaction:

```
Press L>
SELECT + SELECT + LEFT TURN
SELECT + LEAN FORWARD (LEAN FWD)
SELECT + BACK STEP
SELECT + RIGHT TURN
SELECT + BACK STEP
SELECT + LEAN BACKWARD (LEAN BWD)
```

You've just used an undocumented feature to completely change how RoboSapien walks and reacts when walking. Press the Forward button on the remote to make it start walking. When it touches an obstacle, instead of just stopping and saying "Ouch," it karate chops it, backs up, turns away from it, and keeps walking. This was done with 12 quick steps and could have been simplified or customized even more.

There are more than 20 other hidden features and "Easter Eggs" for you to discover and you can log onto **RoboSapienOnline.com** to find more programs and other functions that are not in the manual.

This program will timeout in about five minutes, unless you tell it to keep going by pressing the Forward button twice.

What's in It for Me?

While it's designed for the younger audience, internally, RoboSapien is very different!

After going through the 67 built-in functions, motions, and actions for a couple of hours, I spent the next hour dissecting it. If you look at the detail in some of these pictures, you'll see a very streamlined and interesting internal structure.

One thing you will immediately notice is that, except for the Microphone, everything on the small, efficient main board is socketed. All motors, feedback sensors, and even the power switches are directly plugged into the board and, more importantly, contain detailed labels of their functions and, in the case of the hands, pin configurations.

This made my reassembly of the robot a simpler task, but also makes getting direct access to the hardware a much easier process. A quick glance at the main board told me there was more afoot than just the two spring-loaded, biomorphic feet.

This overview will give you an idea of both the built in hardware and processor.

Main Board:

Internally, the board is barely populated at all. There are only two chips: one small one on the front side for all the socketed connectors and one on the back side for the analog control circuitry.

Both work together with the larger analog processing chip doing all the work of maintaining the joint positions and monitoring feedback from the motors via a "nervous-net" approach to load-bearing feedback, obstructing detection, motor speed optimization, and positioning.

This is combined with some limit switches that detect the extents of motion when triggered by cams

connected to the motors.

How is it possible to get a walking robot with moving arms, hands, fluid actions, and seven motors operating 13 degrees of motion?

This is where you see the wonders of the complicated simplicity in the internal biomorphic design. It starts with the processor specs and the lack of digital control over the robot's motions.

Processors — Digital and Analog, Together!

- 40 second sound ability
- MIDI playback
- 12K firmware
- Four depth stack
- Advanced error handling
- 40 I/Os
- Built-in magnitude comparators
- Stereo DAC outputs
- 3.6V CMOS core with microwatt standby power
- Four programmable counters
- Exceptional supply line noise rejection
- Eight-bit advanced programming set with in-register Boolean and math functions.

RAM

The 29 byte program steps use a programmable action/reaction subroutine function called a "reflex." Like a carbon-based organism, RoboSapien has a reactive reflex that can be programmed for each sensory input. You can program the L> (left), R> (right), or S> (sonic) sensory reflexes as a single six step task, either conditionally or as a direct process. With the built-in routines, the reflex approach expands 29 steps into 84 easy-to-enter steps, optimized for quick recording with the remote.

The amount of steps becomes virtually unlimited when used with IR software control from an external device.

Motors

RoboSapien uses identical, interchangeable motors for each degree of freedom; each has a different cam or connection to move the axis it is installed in. In addition to having the standard electronics for reducing interference with radio or television reception, each motor is wired to act as a feedback device.

The motor load itself attenuates the electrical pulses that feed the power. This self-optimizes the motor's operation in action. The entire system is carefully balanced, so it is not advisable to try to upgrade the motors to faster ones. The motors that are used are extremely efficient — getting 50% efficiency or more while using minimal power. This is one of the reasons the battery life is so substantial.

Operating time for your typical two motor rolling toy that weighs about a pound is in the 30 minute to one hour range. So, how does a heavy robot with seven motors and a far less efficient method of locomotion — walking — last six times that range? The secret is a biomorphic reflex that uses the principles of organic life instead of digital control.

An analog-based system that incorporates spring-like muscles and negating forces lets many of the body parts — like the arms, legs, and feet — have a much lower inertial resistance than they would without being fixed to a point of balance.

The motor then only needs to apply force to one side or another against that point, moving the axis within a fluid range. The natural outcome of this approach is a careful balance that mimics nature's approach to motion. This provides a very flexible and naturally adapted reaction to variances — like terrain shifts and obstacles — and makes the movements more lifelike.

The system goes a step further and uses regenerative energy reclamation. Each motor recycles its energy by generating energy on the return cycle from an action or motion.

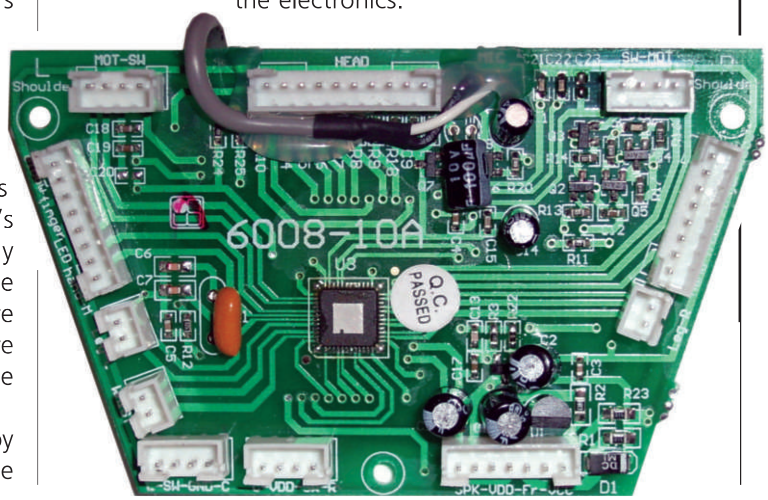
To see this for yourself, try this with your Robosapien:

With the power on, remove both batteries from the robot's foot on the side with the pointed, "sock-grabbing" gripper. Now, manually move the left or right arm up and down. You'll see the eyes pulse and hear staccato sounds from the speaker.

This demonstrates power being generated by the shoulder motors that is being fed back into the electrical "grid," temporarily powering up the electronics. This type of operation feeds back energy that is not used directly by the action you have selected and that's how the runtime can reach its rated six hours.

I've had my Robosapien running for almost 20 hours now — on the first set of batteries — so I think the six hour rating applies to the cheaper batteries.

A warning though: Because of this special battery-extending ability, do not use NiCad or other rechargeable batteries in the robot. They can damage the electronics.



Walk, Don't Run — Unless Programmed To

All of these features come together to make RoboSapien a truly walking robot, not a rolling one, like the inexpensive "biped" robots that were previously available in this price range.

Its walk is another function of the weight and balance of the system. You'll notice that it uses the weight of its body to move and has two walks: one is a kind of hulking waddle and the other is a faster running that uses just the legs to move quickly and does not involve the arms.

While the hardware really is an example of function and form in balance, the software has a few areas that I think many people will see as challenges to master.

First off, all this kicking and swinging is great, but, since it's running on an IR signal, it's hard to finally play a live version of "Rock'em Sock'em Robots." You can actually do this with some programming on each side, but it would be great to have a switchable A-B-C frequency range for IR reception.

Okay, I have to admit it. I have two Robosapiens. I bought one to completely disassemble for this article and, in case my reassembly skills utterly failed and I sat awash in a miserable pool of "I'm not electronically worthy" solder flux, I had the second one. I am glad to say I did not have to take it out of the box until I decided to try a 2004 version of "Rock'em Sock'em Robots" and found the IR controller issues in the process.

The remotes are very directional, so, with some careful positioning, two people can control their bots separately and even shoot an unexpected burp at the other bot on the sly. You can always program them both with your best moves and then let them fight it out like real, autonomous bots!

The second thing that would make Robosapien more interesting right out of the box would be having the autonomous walking mode built-in. I think most people are going to want to plug it in, push the Forward button, and see it start to walk around the house.

It looks to me like this is part of the explorative design, though, meant to encourage owners to experiment with the easy-to-use programming abilities. It takes only a minute or two to program a full walking routine into RoboSapien and a large range of programs will be on the website, from games to — you guessed it — your own RoboRapper.

There are few things I would add to the robot after purchase; to build these things in would make the robot too expensive. It works well as a telepresence device and, with some of the really nice sound effects, I do think Mechanical Minion is a more accurate term. It definitely provides everything the box claims: Attitude, Flexibility, and Incredible Motions.

Hack Attack!

This gives a sense of the overall robot — right down to the grippers — but there are some very special features intentionally built in for anyone wanting to hack-a-bot.

The hands contain points for mounting your own hardware and each element is individually socketed. The grippers themselves are actually interchangeable. I've swapped my left and right grippers and both still work perfectly.

The socket element is one that should not be overlooked. Other than the microphone that is hard-wired for sharp sound responses, everything else — right down to the power switch — is plugged into a socket.

This means your add-on hardware is also just one standard socket away from an interface or addition to the system.

All the sensors are factory-wired for this extra input. The robot is a balanced piece of hardware and electronics, but offers flexibility of action and addition.

The built-in programming steps aren't that extensive, but the hardware lends itself to expansion and easy modification. Programming Robosapien is so easy that it invites anyone with a few parallel pulsing sensors and a screwdriver to be able to teach RoboSapien a new trick.

But Wait, I Told You There's More

We've explored the inner workings of RoboSapien, but it doesn't end there. In part two, we'll get some in-depth insights from the man behind the machine — Mark Tilden — and learn about biomorphic principles, hidden features, and adding IR detection. It's great to see these advanced features in a consumer product like RoboSapien.

I find myself amazed at the array of technology that has gone into this robot and it leaves me wanting to know more about how it was created, the principles of analog control, and applications to other robots and related systems.

Mind Over Machine

This technology isn't new, but it has only recently been combined in this type of biomorphic development. Only recently has a convergence of different backgrounds given us this type of robot.

Until then, my biped is looking pretty guilty and I'm missing a sock. **SV**

RoboBRiX Part 4

by William Benson and Wayne Gramlich

In this fourth installment about RoboBRiX® technology, we are going to discuss the RoboBRiX PICBrain11 hub brix and its microprocessor, the PIC16F876® from Microchip, Inc.

Before we begin, we want to briefly turn our attention to the **MicroBrain8** that we discussed in the previous two articles.

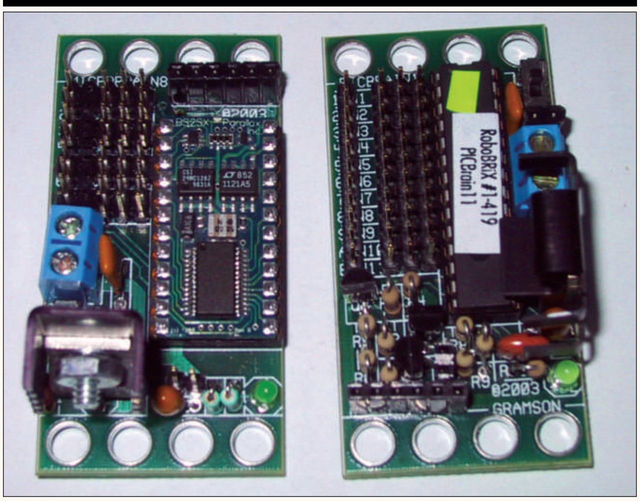
In those articles, each of the robots we described used the Parallax BS2 Stamp as the microcontroller of choice for our **MicroBrain8** hub brix. Parallax, Inc., was kind enough to send us a Javelin Stamp together with a package of Stamps that they call "The Stamp Collection" (consisting of the following Stamps: BS1, BS2, BS2e, BS2sx, and BS2p24) so that we could try each of them with our **MicroBrain8**.

We are pleased to report that all worked equally well with the **MicroBrain8** hub brix, except for the BS1; we didn't test it because of the pin incompatibility. Our many thanks to Parallax for their support!

We also tested the OOPic-C from Savage Innovations and found that it, too, is compatible with the **MicroBrain8**.

Figure 1

MicroBrain8 and PICBrain11 have the same footprint.



RoboBRiX Hubs

So far, we have only discussed the **MicoBrain8** hub in any detail and have said little about the **PICBrain11** and its microprocessor, the PIC16F876. So, in this article, we will talk about this other RoboBRiX hub choice and present a new PIC® compiler — called µCL — that was developed specifically to make programming the PIC16F876 easy to do. To begin, let's compare the **MicroBrain8** and the **PICBrain11** so that we can appreciate their similarities and differences.

Similarities and Differences

Each of these two hubs operates with the same electrical, mechanical (Figure 1), and communications specifications that we discussed in detail in "RoboBRiX, Part 1" in the January 2004 issue of *SERVO Magazine*.

Moreover, all of the commands found in each function brix Programming Table work exactly the same way, regardless of which MicroBrain hub brix you choose. In other words, when looking up to its hub, a function brix would not be able to distinguish the **MicroBrain8** from the **PICBrain11**.

From the user's point of view, the above is also true, but with two important exceptions.

First, the **PICBrain11** can control a total of 11 separate function bricks, whereas the **MicroBrain8** can control just eight.

Second, the **PICBrain11** uses Microchip's PIC16F876 microprocessor, but Microchip does not provide a free BASIC programming environment to its customers.

Programming the PIC16F876

To use a microprocessor, the user typically writes a program using assembly language or one of the higher level and easier-to-use languages — such as BASIC or C — designed specifically for the type of microprocessor being used. Since microprocessors use a special file format called an Intel® hex file, the user must next compile the program to convert it into the required hex file format.

Finally, the hex file must be loaded into the microprocessor chip itself using a programmer or boot loader (provided that one is pre-programmed into the PIC microprocessor).

Microchip does offer a free PIC microprocessor development environment — called MPLAB IDE® – at no cost, but it only supports the more difficult-to-use assembly language. The easier-to-use languages — such as BASIC or C — are available from third party vendors, but only at an additional cost to the user.

Once a program is written and compiled into a hex file, you need a PIC programmer to download the hex file to the microprocessor.

Commercial programmers — like Microchip's PICSTART® Plus — are available for purchase commercially or, of course, you could build your own. Fortunately, there is a simple and free alternative to all of this for RoboBRiX users.

The goal of RoboBRiX technology has been to make the building of robots easier.

In our minds, forcing users to search elsewhere on their own to find a full-featured, but easy-to-use programming environment and then purchase a separate programmer just so that they can use the **PICBrain11** did not seem to be compatible with that goal.

Enter the µCL IDE (Integrated Programming Environment) and the **PICBrain11** Boot Loader.

The µCL Compiler and Integrated Development Environment

µCL stands for Micro Controller Language. It is pronounced like "uncle" without the "n" and rhymes with "chuckle." It is both a stand-alone compiler and an integrated software development environment. The development of this software package was motivated by the evolution of RoboBRiX technology and the goal of making it easier to build robots.

The µCL Compiler

The user's software code must somehow be converted into the Intel Hex file format that is used by the microprocessor itself. The two common ways of doing this are to compile the code or to use an interpreter. We won't go into the differences here, but suffice it to say that a complied language will run an order of magnitude faster than an interpreted language. µCL uses a compiler for this reason to convert the user's code into the Hex format used by the microprocessor.

The µCL IDE

A typical integrated development environment will include: an editor (Figure 2) and a debugger (Figure 3). µCL does not need a linker, since the programs are small enough and the compiler is fast enough to compile all of the code each time a change is made. With this in mind, the µCL IDE incorporates the following primary features:

Targeted for Microcontrollers

Microcontrollers are an interesting subset of processors that differ rather substantially from their more general purpose cousins found in desktop computers. In general, microcontrollers typically use a Harvard architecture where the program code resides in non-volatile, flash, read

Important Features of the PIC 16F876

- 28-pin package
- 20 MHz clock (5,000,00 instructions/sec)
- 14-bit words per instruction
- 8 K of non-volatile flash program memory
- 368 bytes of volatile data memory
- 256 bytes of non-volatile flash data memory
- Eight level hardware call/return stack
- Can reprogram its own flash memory
- One eight-bit timer
- Two 16-bit timers
- Eight channels of 10-bit Analog-to-Digital conversion
- Hardware UART
- Master/slave I2 C support
- Watch dog timer
- Interrupts

Figure 2

Example 1: µCL IDE in Edit Mode.

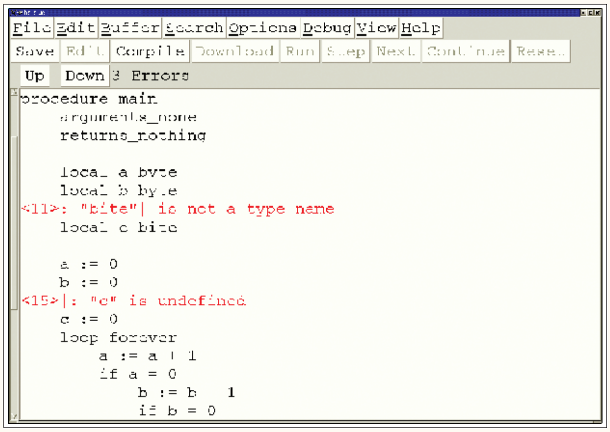


Table 1

V	Print version number of software (i.e., 1.0).
K *n*	Set the socket to n, where n must be between 1 and A, inclusive.
I	Talk to the module on the socket specified by the K command (see above).
C *aa*	Perform a clock adjust on the selected socket; **aa** specified as 00.
P pp	Show the page of program memory at pp00h to ppFFh.
E rrrr cc	Show the cc bytes of data from file registers, starting at rrrr.
S rrrr vv	Set register rrrr to vv.
G aaaa	Transfer control to address aaaa.
X	Transfer control to address 0008.
R cc	Send byte cc to the currently selected module; receive two byte response.

only memory and the data reside in volatile random access memory.

In addition, the individual port pins of a microcontroller can be easily programmed for specific uses.

Microcontroller Neutral

µCL's set of instructions is written to make intuitive sense to the user, just as BASIC has always done. The present µCL code generator is geared specifically towards the popular PIC microcontrollers from MicroChip, but the set of instructions available to the user would remain — with few exceptions — unchanged if a new code generator was written to support one of the other microcontroller families sold by different vendors.

Platform Neutral

The µCL compiler runs on multiple platforms — Linux®, Windows®, and — eventually — MacOS®.

Libraries

Quite often, a body of code is general enough in nature that it can be used without modification in an application different than the one for which it was originally written. In µCL, users can save this kind of code in a library and make it accessible — when appropriate — to new applications that they are writing.

For example, a user might build a table of commands taken from the **Servo4** Programming Tables and save it as a Library to be used whenever writing code for the **Servo4** brix. Libraries can be shared between users.

Debugger and Screen Editor

The µCL IDE has a fully integrated editor and debugger. The debugger supports user-set breakpoints: line by line, single stepping, variable inspection, and variable updating.

Every attempt has been made to make the µCL language friendly for beginning users. For example, error messages are localized to the line where they occur and are written in a style that is concise, but comprehensible to the user.

Different from C or C++ languages, µCL has much of the intuitive feel that has made BASIC languages so popular, but it contains a much richer and more powerful set of instructions than found in ordinary BASIC compilers.

Moreover, µCL has eliminated the use of braces to identify nested statements, with the result that the tedious task of dealing with mismatched brace errors is eliminated, as well.

The BASIC compiler that probably compares best to µCL is PBasic 2.5, available free of charge from Parallax. Certainly, compared to the C compilers presently available commercially or in freeware, µCL ranks high on the scale of intuitive ease and functionality.

Cost to User

It's *free*! Visit the µCL website shown at the end of this paragraph. There, you can page through the user's manual for a description of the language specifications and download both the manual and the µCL IDE at no cost.

http://Gramlich.Net/projects/ucl/v1/index.html

The PICBrain11 Boot Loader

There is no need to purchase a separate PIC programmer in order to use the **PICBrain11** hub brix. Every **PICBrain11** comes with its own MicroChip PIC16F876, pre-programmed with a boot loader. The built-in boot loader lets the user download a compiled code directly to the PIC16F876 without using a programmer. The download process functions in much the same manner that BASIC programs are downloaded to the Stamp in a **MicroBrain8**. Swapping the PIC16F876 from the **PICBrain11** to a chip programmer and back again is a thing of the past.

Using the boot loader is an easy, three step process:

1. First, connect the programming cable between your laptop (or desktop) computer and the programming port on the **PICBrain11**.

2. Next, load the terminal emulator that was included with the original operating system installation. For Windows users, it will be HyperTerminal and, for Linux users, it will be minicom (but, of course, almost any terminal emulator program will work just as well).

3. Finally, using the appropriate boot loader commands (see Table 1), download the program hex file to the PIC 16F876 and you're all done.

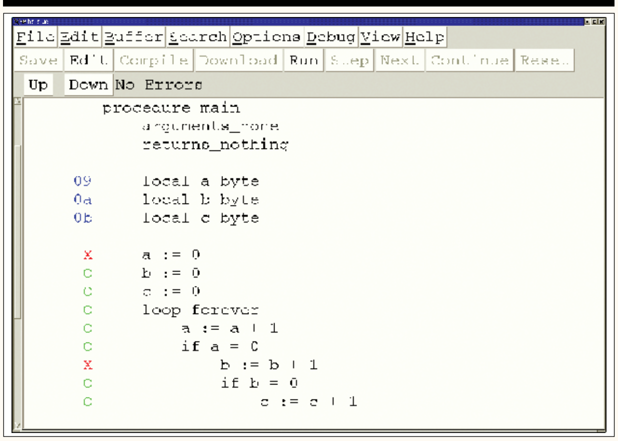
Conclusion

If there is one message that we want to get across, it is that the goal of RoboBRiX technology is to make building robots easier. We also recognize that having the proper software tools to do this is just as critical as having the proper hardware. So, along with making it easier to build robots, we have endeavored to make it affordable, as well. The µCL IDE and the **PICBrain11** boot loader are the results. They provide a free, but powerful, software development tool and an easy, cost effective way of programming the PIC 16F876.

Next month, we will introduce a new function brix designed to provide odometry information. This is the first step on the path to developing the ability for our robot to navigate its environment with a reasonable degree of precision. **SV**

Figure 3

Example 2: µCL IDE in Debug Mode.



Our resident expert on all things robotic is merely an Email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Pete Miles

Q. Could you give me a list of fairly inexpensive microcontrollers with their corresponding compilers and programmers? I am currently using a BS2, but they are expensive to buy in quantity. I have also looked into PIC microcontrollers because of their low price per controller. Unfortunately, the compilers and programmers I need would be over $100.00. What do you recommend?

—Bryan Hood
Via Internet

A. It's difficult to recommend one microcontroller over another because — for the most part — the microcontroller(s) you end up using are usually selected based on their costs and personal taste. Personally, my first preference is to use what I already know how to use or something that is similar, so that I can minimize the learning curve for the new device. If that doesn't work, then I try to look at the long-term benefits and costs of choosing a new microcontroller.

As you probably have determined by now, I am a BASIC Stamp fan. They are a fine product and can do about 95% of the things I want them to do. When I started looking around for another microcontroller because the application I was working on did not work on the BASIC Stamp, cost was my primary driver. I, too, was amazed at the costs of programmers and compiler languages.

Since I wanted to keep my costs to a minimum, I decided to learn how to program in assembly language for PIC microcontrollers. The reasons for this decision were: the language is free (from Microchip), the microcontroller chips are cheap, there is a wide variety of them to choose from, and the BASIC Stamp is based on the PIC 16C56.

Since I liked the BASIC Stamps and the customer support, I decided to get the PIC programmer and learn Parallax assembly. Parallax sold their PIC programming development tools to Tech Tools (**www.tech-tool.com**). It took awhile, but I eventually figured out how to program in Parallax assembly language. Later, it turned out that, going this route, I ended up learning the same assembly language for the Scenix microcontrollers that are now at the heart of the new generation of BASIC Stamps. After awhile, I wanted to program some of the eight-pin PICs, but I needed an adapter for my programmer.

Because of the costs of the adapter, I decided to get the PicStart Plus from Microchip, since it can handle more chip configurations than the programmer Parallax had at the time. I also decided to learn the regular assembly language from Microchip (Parallax assembly is very similar to Microchip assembly). Eventually, I obtained the MBasic compiler and development package from Basic Micro (**www.basicmicro.com**) because I still like to program in BASIC. To me, assembly language programming is still time-consuming and MBasic is compatible with the BASIC Stamp language.

This is kind of a roundabout story, but it does illustrate what many people go through when working with microcontrollers. They start with one product, look for low cost alternatives, explore those alternatives, and eventually end up investing in the compiled language development tools they tried to avoid in the first place. In some cases, they go back to the original microcontroller they started with.

Some of the language compilers and programmers available today may seem expensive right now. In the long run, though, they tend to be one of your best investments — if you don't end up spending a lot of your time and money exploring too many "low cost" options.

The core microcontrollers inside BASIC Stamps are the Microchip PIC 16C56s for the BASIC Stamps 1 and 2 and the Scenix (**www.ubicom.com**) SX28ACs and SX48ACs for the other BASIC Stamps. Microchip has an extremely wide selection of microchip options to choose from and the SX chips are extremely fast (up to 75 million instructions per second). The Flash programmable microcontrollers are gaining a lot of popularity for low cost solutions.

For low cost options for programming the SX and PIC microcontrollers, I would recommend learning assembly. For the SX products, I would recommend the SX-Key or the SX Tech Tool Kit (which includes the SX-Key) from Parallax.

For the PIC Microcontrollers, I would recommend the PicStart Plus or the PicKit 1 Flash Starter Kit from Microchip. All four of these products include the assembly language development software and a programmer. The PicKit is limited to eight and 14 pin, Flash programmable microcontrollers, but it includes a limited edition of Hi-Tech PICC C compiler (**www.hitech.com.au**).

If you are interested in working with both SX and PIC microcontrollers, then take a look at getting the assembly language and programmer from Tech-Tools. The two assembly languages are very similar.

If you are looking for a Basic compiler, then I would recommend that you look at the MBasic compilers from Basic Micro and PicBasic compiles from microEngineering Labs (**www.melabs.com**). Both of these compilers use a Basic language syntax that is compatible with the BASIC Stamps. Thus, your old programs can be programmed into dedicated microcontrollers. If you are planning on using one of these compilers, I would recommend that you use the programmers these companies sell, since they have been shown to work. Note, these compilers currently do not support the SX microcontrollers.

There are a lot of microcontroller programmer designs available on the Internet, but I would recommend that you purchase one from the company that developed the programming environment. I have read many Email posts from frustrated people who are trying to build their own programmer from a design found on the Internet. In the end, many of them eventually ended up using a commercially-built programmer.

There is another microcontroller option that you should take a serious look at — the AVR microcontrollers from Atmel (**www.atmel.com**). They are gaining a lot of popularity across the robotics community. The best place to get started with the AVR is the STK500 starter kit. This kit includes a full programming environment that uses Atmel's assembly language and a programmer with push buttons and LEDs for testing purposes.

The individual microcontroller prices are comparable to the PIC and SX microcontrollers. The popular BASCOM Basic compiler from MSC Electronics(**www.mcselec.com/bascom-avr.htm**) is very powerful; it is one of the lowest cost Basic compilers available for any microcontroller. When considering the costs for getting started in programming microcontrollers with a Basic compiler, the AVR microcontrollers might be your most economical choice.

Q. I am currently working on a bot that is based on an old Mac SE that runs on System 6. I want to be able to keep the monitor for the face — á la 790 from Lexx. One question I have is how to power the PC using a car battery. Second, where can I find information on writing code for System 6 to control the bot through the parallel port? Also, would the Mac interface with RoboBRiX?

— Dan Askew
Via Internet

A. To power that old Mac with a car battery, you are going to need a 12 V DC to 120 V AC power inverter. These power inverters are actually very common and they can be found at most auto parts and sporting goods stores. They can plug directly into the cigarette lighter port in your car or directly to the 12 volt car battery and they come in many different power ratings — from 100 watts to several kilowatts. Keep in mind that, the higher the power rating, the more expensive they become.

All you have to do is determine how many watts the computer needs to run and choose a power inverter with a watt rating that is greater than what you need. Xantrex (**www.xantrex.com**) is one of the leading manufacturers of these power inverters. Their website has a great deal of valuable information about these products. Follow the automotive link for the power inverters that you will need. Their downloadable manuals provide good information on how to use and size them and estimate how long your battery will last.

In computer dating terms (similar to Carbon-14 Dating), the Mac SE is ancient. You are going to have to poke around on the Internet to find information/people to help you program that machine of yours. One place that can get you started in the right direction is **www.faqs.org/faqs/macintosh/programming-faq/**

RoboBRiX use a serial communication protocol with one start bit, eight data bits, no parity bit, one stop bit, and a speed of 2,400 baud (bits per second). Though not impossible, it is not easy to make a parallel port act like a serial port. It may be easier to use the Mac's serial port to communicate with the RoboBRiX's PICBrain11 or to have the MicroBrain8 communicate with multiple RoboBRiX modules. You will need to write a multiplexing program for the PICBrain11 or the MicroBrain8 that will take the single input from the Mac SE to control multiple modules.

Before doing this, you are going to need to convert the RS422 serial communication protocol that the Mac uses into either RS232 or 0-5 V TTL voltage level signals. The PICBrain11, MicroBrain8, and the Servo4 modules can accept both standard RS232 voltage level signals and 0-5V TTL voltage level serial communications, whereas the other RoboBRiX modules only use 0-5V TTL voltage level signals for serial communications.

The Mac SE. Photo courtesy of Nerraux, **www.ThaMike.com**

A great tutorial about interfacing with the Mac's serial communication port can be found at **www.rdrop.com/~cary/html/serialportdocs.html/** There, you will find pinout descriptions to the eight pin DIN connector that the Mac uses and wiring configurations and instructions for the RS232.

Detailed information about the RoboBRiX's specifications can be found at **www.robobrix.com** and **http://gramlich.net/projects/robobricks/index.html** Make sure that the RoboBRiX's modules are electrically compatible with the Mac before wiring all of the components together.

This sounds like a fun project using an old Mac as the main robot controller. When you get done with it, you should send in some photos for *SERVO Magazine*'s "Menagerie." I'd bet a lot of people will love to see new uses for their old Mac computers.

Q. I need some thoughts and input on belt drives verses chain drives. The drive would be coming off the motor at about 2,000 RPM (max) and going to a shaft at about 200 RPM. The motor is DC at about 2 HP. Which is best, considering: original cost, efficiency, noise, dependability, upkeep, etc.? What would be the minimum size that I could get away with?

— Bernie
Via Internet

A. Both timing belts and roller chains will work just fine for reducing the speed from 2,000 to 200 RPM in your 2 HP drive system. Per your request, I compiled Table 1, which shows a relative comparison between the two types of drive systems. Choosing one approach over the other is more of a personal preference, since they both will work. I would personally go with timing belts when I am looking for a high coefficient of coolness and roller chains when I know I will be taking the system apart many times. For this application, however, it is really a personal preference.

You are going to need a 10:1 speed reduction in this system. Based on the minimum specs shown in this table, the speed reduction cannot be achieved with a single stage (one drive pulley — one driven pulley) using standard available components, but you can use a larger pitch size that will allow you to use smaller diameter pulleys on the 2,000 RPM shaft and will help you reduce the number of stages needed to achieve the 10:1 speed reduction.

There are several PDF files at Martin Sprocket and Gear, Inc. (**www.martinsprocket.com**), that provide instructions for properly sizing sprockets and timing belts for your application. Follow the Power Transmission Components link on the main page and choose the type of component (i.e., sprockets or timing pulleys). Then go to the Publications link. There, you will find several PDF files that describe how to size the components. This will be good information to have in your design library.

	Roller Chain	**Timing Belts**
Original Cost	Low	High
Efficiency	Good	Good
Noise	Loud	Quiet
Dependability	Longer	Shorter
Upkeep/Maintenance	Higher	Lower
Reliability	Good	Good
Assembly	Easy	Harder (requires belt tensioning and/or removing pulleys)
Maximum Surface Speed ($2\pi *$ radius$*$RPM)	3,000 ft/min	16,000 ft/min
Life	15,000 hours	1,000 hours
Minimum Recommended Pitch Size	"No. 25" 1/4" Pitch	"L" 3/8" Pitch
Minimum Pulley Diameter (Based on Minimum Pitch)	2.13", 25 teeth	1.67", 14 teeth
Minimum Belt Width	N/A	1/2" inch, typical

Table 1. Relative comparison between roller chain and timing belt drive systems.

Q. I have been a mainframe programmer since 1971 (RPG, mainframe assembler, COBOL, CICS, BASIC, etc.), but I never did learn PC assembler. I just subscribed to *SERVO Magazine*. I want to get into robotics programming. I don't want to waste time learning another "flash in the pan" programming language that will fade away. Which language would you recommend for present and future robotics?

— Jerry Nicholson
Via Internet

A. Based on these criteria, probably the best two computer languages that are currently being used — and will be around for a long time — are BASIC and C. It sounds like you may already know both of these. There are BASIC and C languages/compilers/interpreters for most of the microcontrollers available today and, when a new microcontroller comes out, new BASIC and C compilers soon follow.

Keep in mind that there will be subtle variations of these languages from one microcontroller manufacturer to another and from one software development company to another, but they are very similar so that migrating between them is a fairly straightforward process. The main difference that you will notice will be the commands that access the specific hardware features/functions of the microcontroller. The BASIC and C programming languages are probably your best bet for a long-term, stable programming environment. SV


SERVO Magazine Challenges You To
TETSUJIN 2004
Man joins Machine in a test of Imagination Physical Strength and Mental Agility
October 21-23
Santa Clara, CA
The Japanese call it Tetsujin — Iron Man. This groundbreaking competition combines both brains and brawn. Tetsujin 2004 presents the ultimate challenge: Produce a powered, articulated exoskeleton to assist a human operator in lifting weights. Teams will compete against the clock and each other in this robotic showdown!
Held at the Santa Clara Convention Center, this world class competition — sponsored by SERVO Magazine — spotlights this emerging technology in the robotics industry, which has applications ranging from national defense improvements to new mobility for the disabled.
Teams for the competition are welcome from all sectors. Now is the time to gather your friends and co-workers and begin imagining. Answer the challenge of Tetsujin 2004! The earlier your team registers for the competition, the longer you will have to create and develop your entry.
Visit the SERVO Magazine website, click on the Tetsujin link, and sign up today.
Complete rules, registration forms, and FAQ are available.
www.servomagazine.com
Enter the competition today!

Motion Control ... ¢HEAP

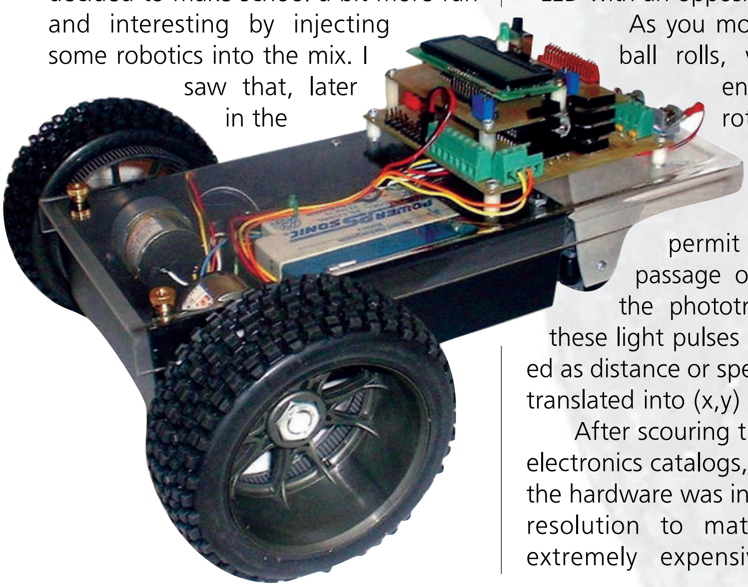
by Rich Kappmeier

Would you like to build inexpensive shaft encoders with good resolution for around $6.00 each? How about a simple proportional control program that will have your robot moving and exploring with a purpose?

Motion control is an important aspect of robotics. Unfortunately, the hardware can be expensive and the software can be difficult to implement. With this article, I will demonstrate how to make some inexpensive shaft encoders and provide a simple way to implement basic motion control using a proportional algorithm.

D-I-Y Makes It C-H-E-A-P

Several months ago, on the first day of my industrial controls class, I found out about a final project that was due in the last week of the term. I decided to make school a bit more fun and interesting by injecting some robotics into the mix. I saw that, later in the semester, we were going to cover proportional feedback loops. This got me thinking about my robot — Polythene Pam — and its lack of odometric know-how.



However, if Pam is going to do any odometry, it will need a way to determine the distance it has traveled and the relative speed of its drive wheels and I will have to implement this as inexpensively as possible. That means shaft encoders.

A shaft encoder is a feedback device that allows a system to monitor the rotational motion of a shaft. Generally, optical encoders use a beam interrupt scheme. The most common sort of optical encoder you will encounter is inside a computer mouse. Each optical mouse has two encoders — one for the x-axis and one for the y-axis. Those individual mechanisms consist of a slotted disk with a shaft that contacts the mouse ball and an LED with an opposing phototransistor.

As you move the mouse, the ball rolls, which rotates the encoder shafts, which rotates the slotted disks, which causes the slots in the disks to alternately permit and prevent the passage of the LED light to the phototransistors. Each of these light pulses can then be counted as distance or speed by the CPU and translated into (x,y) pointer motion.

After scouring the Internet and my electronics catalogs, I found that either the hardware was inexpensive — with a resolution to match — or it was extremely expensive ($200.00+ in many instances) with more resolution than I needed.

I hacked some equipment I grabbed from the scrap bin at work, but I was only able to get a 10° resolution. This was not even close to what I wanted. If I was looking for an economical near 1° resolution, I was going to have to make my own encoders.

While laying out a circuit board at home, I realized that, if I could print stripes on the outer rim of a clear plastic lid or container, I could use that as the encoder disk! That encoder disk — coupled with a slotted optical switch — would make a handy and reasonably priced shaft encoder.

Rapid Considerations

There are some limitations that need to be addressed before anything else. First, how fast will the encoders be rotating? Second, what resolution will work out best for my application? It turns out that both of these questions are related.

Rotational speed is important because the faster the encoder wheel zips through the optical switch, the shorter the amount of time the processor has to register any changes. If the velocity of the rim and the resolution of the encoder stripe are both high, the processor may not be fast enough to catch all of the transitions. To illustrate:

Disk revolutions per second = 3
Disk resolution = 1°
(360 transitions per revolution)

(3 rev/sec) * (360 transitions) = 1,080

transitions per second
(1 second) / (1,080 transitions) = each transition is 925 µs long

A 925 µs transition time may seem long in the world of microprocessors, but, if you have other subroutines running simultaneously — like loading display data to an LCD or reading the other sensors — transitions may get dropped. Dropped transitions will cause large cumulative errors and throw the robot off course. A balance between speed and resolution must be struck.

The size of the encoder wheel really only matters in regard to the printable resolution and the width of the beam to be interrupted. If you have access to an excellent printer and copier, feel free to make the wheel as small as possible. The rest of us will need to think bigger. In addition, the beam must also be completely blocked by the interrupter. I felt a safe minimum dark segment width of twice the beam width would be satisfactory. With large encoder disks, neither of these issues will be a problem.

A good natural resolution to start with on your robot might be 256 segments per encoder, which is sometimes referred to in binary degrees. This yields about a 1.4° resolution and also allows for a single revolution to be counted within a byte of memory.

I wanted to cut costs as much as possible, so I tried to include only the minimum hardware necessary. After all, the textbook for the class was $120.00 by itself! I figured code is free and textbooks aren't, so this left me with the optical switches, the encoder wheels, the comparator, and the microprocessor (which was already onboard).

For the slotted switches, I used the OPB818 manufactured by OPTEK Technology. I also decided that I wanted a definite ON/OFF boundary for the switch output, so I added an LM393 dual comparator to the circuit. I didn't want any half-shadow, mid-voltage signals messing up what the processor was trying to read. With a comparator, a definite threshold can be set so that all signals below the threshold — in my case, 2.5 volts — will be a digital low and all signals above will be a digital high.

You may have noticed that I am incrementing the count every time the signal changes. This allows me to count each segment — dark or light — and yields greater resolution, but counting the signal changes requires the microprocessor to remember what the last state was and then compare the current state to that memory.

Other systems may only count the dark to light (low to high) or light to dark (high to low) edges. This means that it takes a dark and light segment pair to increment the count and effectively halves the possible resolution. By counting using only a low to high edge-triggered counter, you would lose the high to low edges. These chips just don't see those high to lows; they're not set up for it.

The Not So Hard Software

As mentioned before, the software utilizes a proportional algorithm to adjust the speed of one wheel (the left for Pam) to the speed of the other. Proportional control is very easy, if it is taken in steps. Let's define some terms first:

Set Point (SP): The desired output.

Process Variable (PV): The signal

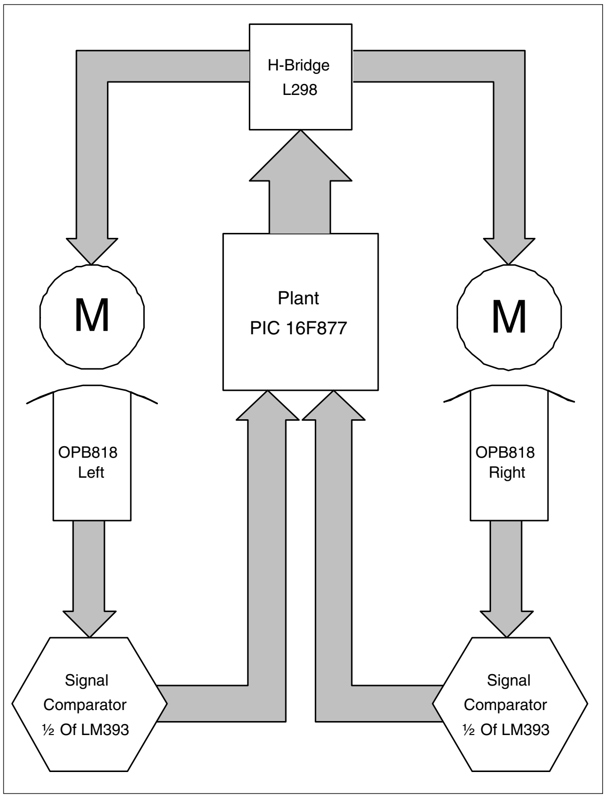
Three Examples of Proportional Control

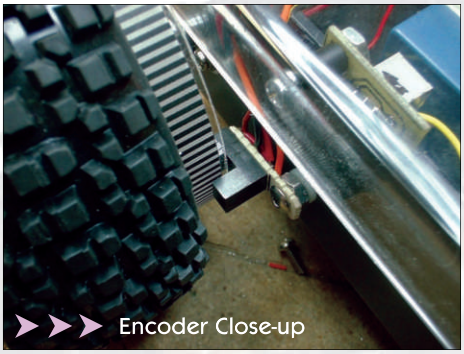
Left Motor Count (PV): 560 — **Right Motor Count (SP): 563**
The Left Motor is too slow!
563 – 560 = Error of 3
(3 * 60) + 700 = 880 for the Left Motor PWM Output
Left Motor speeds to catch up.

Left Motor Count (PV): 821 — **Right Motor Count (SP): 816**
The Left Motor is too fast!
816 – 821 = Error of -5
(-5 * 60) + 700 = 400 for the Left Motor PWM Output
Left Motor slows!

Left Motor Count (PV): 995 — **Right Motor Count (SP): 955**
The speeds are equal!
955 – 955 = Error of 0
(0 * 60) + 700 = 700 for the Left Motor PWM Output
Left Motor maintains speed!

Figure 1. A functional diagram.




Encoder Close-up

from the feedback sensor.

Error: The difference between the Set Point and the Process Variable.

Gain: A number used to adjust or amplify an error signal. It's really only a term that scales an error to an output signal.

PWMcon: A constant in my software that equals 700. This is an arbitrary variable name I have used to define the Pulse Width Modulation (PWM) output settings for both of Pam's PWM modules on its PIC 16F877 microprocessor.

In regard to the system we are setting up here, the encoder count of the right wheel is the SP and the encoder count of the left wheel is the PV. As these counts are determined, there is some math going on. First, we have to determine the Error:

Error = SP – PV
or in Pam's case:
Error = (right count) – (left count)

Normally, an SP is exactly that — a set, desired output, like the temperature on a thermostat. In this application, the SP of the right wheel is static only for a short time because the wheel keeps rotating. The speed — or rate of change — of the right wheel is really what the SP is, not the actual count itself.

Once the error is calculated, it can be plugged into the proportional equation:

Output = Error * Gain
or in Pam's case
Left Motor Output = (Error * Gain) + PWMcon
while
Right Motor Output = PWMcon

As you can see, we are adjusting the PWMcon output of the left motor while leaving the PWMcon of the right motor alone.

The gain factor is a term that is arbitrarily set by the programmer. A good way to set the gain is to start low — even as low as two — while working your way up the scale and observing the results. With Pam's system, I started down at 20 and observed the robot as it went through a few program runs.

At 20, it wobbled slightly as it began to move — not enough response. I increased the gain by increments of 10 and, at 60, it smoothed out. I kept rolling up by factors of 10 to see what would happen and, at 90, it began to wobble again — too much response. A gain of 60 turned out to do the trick.

What does all of this mean to the motor outputs? If, as the wheels of the robot are turning, you place a finger

How to Make Your Encoder Disks

Find two clear, plastic containers that are as close to the diameter of your drive wheels as possible without being obstructive.

Cut the bottom of the containers down to the width you would like and carefully smooth the edges with a file or sandpaper. I cut mine down to 1 cm wide with a cutting disk and my drill press (Photo 1).

Mill a hole into the encoders for the drive shaft. I used my Dremel tool and a milling bit at maximum RPMs (Photo 2). I did try to actually drill a hole first — starting small at 1/8" — but, once I got up to the 3/8" bit (the minimum size I needed), the plastic shattered. Oooops!

Create two interrupter patterns and print them onto a clear label sheet. Here's how I did mine in Paint Shop Pro (similar to Photo Shop).

a. Start a New drawing.
b. Make it a black and white (two bit) color scheme.
c. Set the pixels per inch to 256.
d. Set the drawing dimensions to 1" by 0.1".
e. Stripe every other row of pixels and the long edges. It will look like a black and white ladder with one white end and one black end.
f. Find the exact circumference of your encoder wheel.
g. Resize the drawing to fit the circumference and width of the disk.
h. Do a Print Preview to make sure it's going to print properly.
i. Print it.
j. Place a sheet of Avery #8665 clear plastic label material into a copy machine. Make sure it will print on the correct side!
k. Copy the interrupter patterns onto the label material. When I copied mine, I selected Transparency for the paper type and darkened the toner slightly.
l. Cut out the interrupter patterns and stick them to the rims of the disks.

You may be asking why I didn't just print directly onto the label material. I tried this at first and found that the optical switch doesn't see the ink from my Bubble Jet printer at all. I'm guessing that it's not dense enough to block the light.

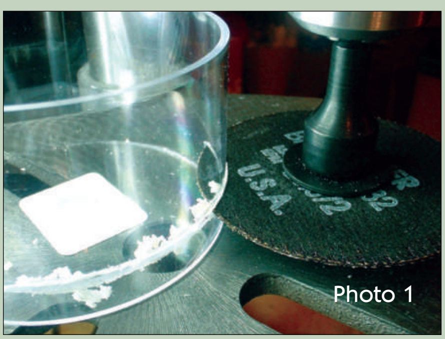


Photo 1

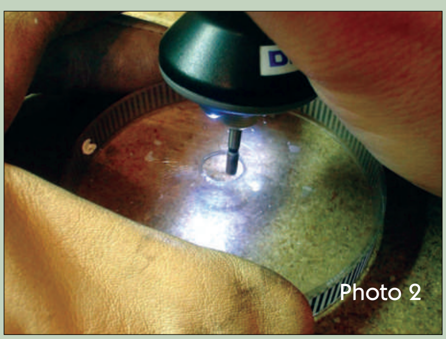


Photo 2

lightly on the right wheel, it slows. This creates a negative error in the system because the left wheel is now moving faster than the right. As this error increases, the software will compensate by applying the math and decreasing the power via a reduced PWM signal to the left motor, slowing it.

Ultimately, you should be able to grab the right wheel and stop it completely (Watch your locked rotor current!) and see the left wheel come to a stop.

Now, if you drag the left wheel with your finger, something different happens. This new error (in the positive) grows because the left wheel is now moving slower than the right. As it grows, the proportional control kicks in and increases the power to the left motor in an attempt to maintain speed and keep up with the right. With Pam's motors, I can hear a definite laboring sound in the gearbox of the left motor.

In a Perfect System ...

Proportional control systems do have their problems — primarily overshoot — and may not be a sufficient solution in every control application. Overshoot and its evil twin — undershoot — are usually the cause of a poorly chosen gain factor. Having too much gain will cause the output signal to oscillate above (overshoot) and below (undershoot) the desired SP and the system will never quite settle. Too little gain and the system won't respond efficiently. A correctly chosen gain will help alleviate many potential problems.

Fractional feedback signals are a minor problem, as well. You will notice that this system cannot compensate for errors smaller than one segment. Any errors in the system will only register as whole numbers.

Another issue I was forced to contend with was the size of the PWM registers for the 16F877 microcontroller. These are 10-bit registers with a maximum setting of 1,023. In my program, I only use the first 1,000 counts, ignoring the final 23, since they didn't make much difference in the power output, anyway.

I found out the hard way that — if an error gets too large in either direction — it causes the PWM registers to overflow or underflow. With a PWMcon equal to 700 and a gain of 60, an error of more than +5 will overflow the left register above 1,000, causing it to wraparound all the way back down to the bottom of the scale. If the error was, for example, +6, the register would overflow down to 37, which is equal to 3.7% of the total available power because

$$(6 * 60) + 700 = 1060$$

Therefore, 1,060 is 37 counts greater than the 1,023 limit of that 10-bit register. The motor will slow down instead of speeding up!

Errors of less than -11 will underflow the register below 0 and all the way back up to the top of the scale. The motor will speed up instead of slowing, throwing the system far off course! I built the limits of 5 and -11 into the software to prevent any over/underflow problems.

Parts List

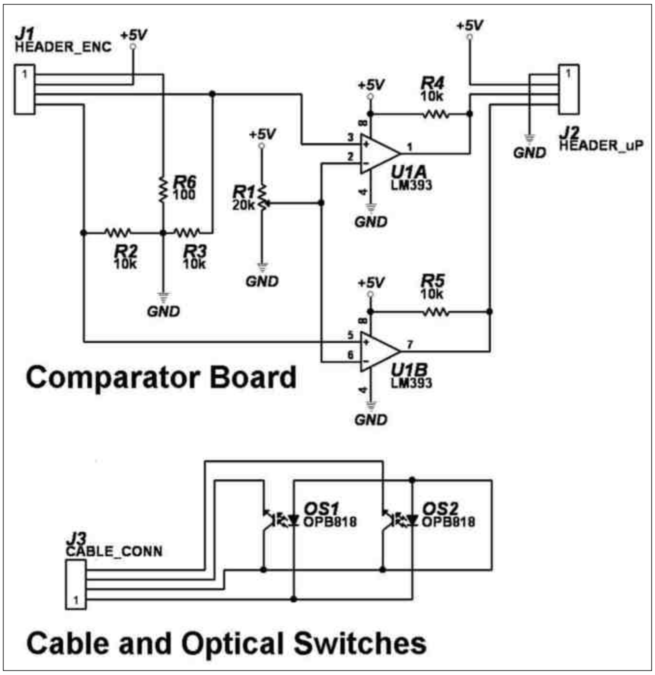
Item	Quantity	Cost Each	Total Cost	Supplier
Clear plastic container	2	$1.99	$3.98	The Container Store
OPB818 optical switches	2	$2.37	$4.74	Digi-Key
LM393 comparator	1	$0.55	$0.55	Digi-Key
Sheet of Avery #8665 clear label stock	1*	$1.00	$1.00	Office Depot
Discrete components; scrap of perfboard	few	$1.00	$1.00	My component bins
		Total for All Parts	$11.27	

* A package of 20 Avery #8665 sheets costs about $20.00 new. I already had these.

On Purpose

There are few things in robotics that look cooler than a machine that follows commands smoothly and efficiently, moving with a purpose. Properly implemented motion control can change a wandering curiosity into a tool of exploration. With the inexpensive do-it-yourself encoders and basic motion control outlined in this article, you will be able to expand and refine these techniques and change your robot into an explorer with a purpose. SV

Figure 2. A cable and comparator board schematic.



Book Review

by Edward Driscoll, Jr.

GEARHEADS

The Turbulent Rise of Robotic Sports

TV shows like *Battlebots* and *Robot Wars* would seem to be a politically correct network programmer's dream: lots of violence to placate bleary-eyed cable TV viewers home from a hard day at work, but no guns, no blood, and no people getting hurt. This makes it rather ironic that, when the DIY Network was assembling their recent *Robot Rivals* TV show, they wanted the robots to compete with each other by performing specialized tasks instead of smashing each other up (see the *Amateur Robotics Supplement #2*, still available as a back issue; **www.servomagazine.com**).

About This Book ...

In *Gearheads, Newsweek* technology correspondent Brad Stone examines the history of robotic sports, from their cultish early years at universities and sci-fi conventions to today's televised extravaganzas — and the turmoil that threatened the whole enterprise almost from the beginning.

By turns a lively historical narrative, a legal thriller, and an exploration of a cultural and technological phenomenon, *Gearheads* is a funny and fascinating look at the sport of the future today.

As someone associated with the show told me, "I'm not keen on these programs that have people building robots that bash each other and try to beat each other up.

That's contrary to Asimov's laws of robotics!" Of course, Asimov's laws — particularly his third law — were more concerned with a robot protecting its own existence rather than fighting another robot, but the point is obvious: no matter how "made for TV" fighting robots are, there will always be a contingent of people who are against it and want to see robots on TV in a more benevolent role.

From Guerrilla Theater to International TV

This dichotomy is one of the themes that Brad Stone — a *Newsweek* technology columnist — explores in *Gearheads: The Turbulent Rise of Robotic Sports* (Simon & Schuster, 2003). Stone traces the evolution of robotic sports from its counterculture origins in a Berkley street theater guerrilla performing arts show to its big-time popularity on American and British TV.

Along the way, he explores the two chief factions in the sport: those who see it as great TV and those who see it as a negative reflection on our culture and the place of robots in it.

Of the former, Stone describes how TV shows such as *Robot Wars* and *Battlebots* became surprise hits on cable TV. Stone goes into a bit of detail about who pushed for the shows and what behind-the-scenes maneuvering went on behind closed doors to get them on the air.

Looking at the detractors, he describes how Segway inventor Dean Kamen, among others, was vehemently opposed to these TV programs, despite their showcase for new technologies.

Stone also does a thorough job of highlighting the other battles the robot warriors fought — in the courtroom.

However — whether you're a supporter or opponent of robotic battles — if you're looking for real "gearhead" information about actual robot construction, this book won't be all that satisfying.

It's definitely more focused on the back room negotiations and maneuvers that brought the shows to your cable box and I would have preferred that the book had been indexed, enabling readers to quickly look up the robots, their builders, and the other behind-the-scenes people mentioned throughout the book.

Birth of a Sport

All in all, the book does a good job of describing the spectacular growth of robotic sports in the 1990s.

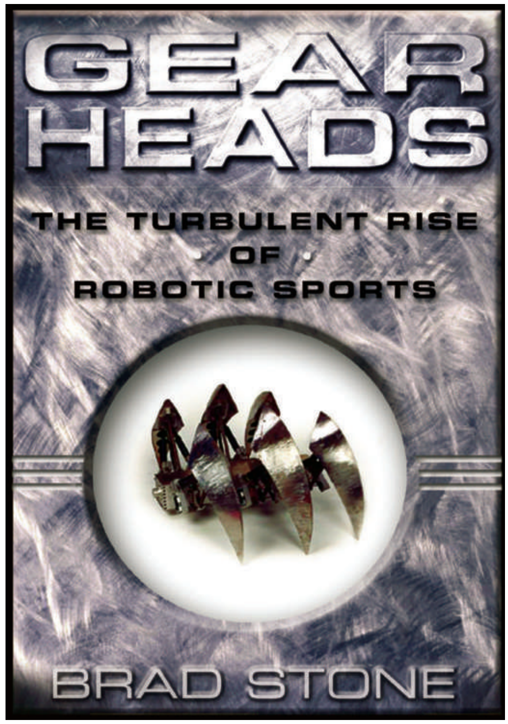
While the dot.com industry's booms, busts, and consolidations have gotten the lion's share of ink, other technologies grew rapidly during that same decade — not the least of which were robotics and cable TV programming — and deserve to be documented, as well.

Don't forget to look for Dan Danknick, *SERVO's* intrepid editor, who makes a few appearances in *Gearheads*.

It's too soon to predict the long-term viability of robot-bashing shows. It's entirely possible that they're just another flash-in-the-pan moment — a media-driven fad on par with skateboards, CB radio, and rollerblades.

All of those hobbies — although begun as fads that received enormous amounts of initial publicity — still maintain serious practitioners to this day. For many, many young boys (and, to be fair, a large number of girls, as well), the idea of robots bashing each others' circuits in is practically hard wired in at birth — just look at how many sets of Rock'em Sock'em Robots have sold over the past 40 years or so!

So, whether or not they get international TV coverage, robots will be fighting each other for a long time to come, no matter how un-PC it may seem to some naysayers. **SV**



Everything You Ever Wanted to Know About Binary Numbers, But Were Afraid to Ask

by Jack Buffington

Recently, there was a joke going around that stated, "There are 10 types of people in this world: those who understand binary and those who don't." For those of you who are now scratching your heads in confusion, this article is for you. The binary number system is the basis of nearly every computing device in existence today. Knowing how it works can be an asset when designing software for processors, such as those a robot might use.

Certainly, everyone reading this article knows the decimal number system inside and out. It is the number system that we use in our daily life that consists of the numbers 0 through 9. We could say that the decimal number system has a radix of 10, since there are 10 numbers used. When counting in the decimal system, we use increasing values until we run out at 9. At that point, we add a second digit to help represent the next value. At 99, we can no longer increment these digits, so, once again, we add another digit to get 100.

The binary number system has a radix of two. It only has two numbers: 0 and 1.

0
1
10
11
100
101
110
111
1000

Counting in binary is the same as in the decimal system, except that, since there are only two different numbers to use, we run out more often. The table on the left shows how you would count to 1000 in binary, which represents 8 in the decimal number system.

Now you know how to count in binary. What if you are given a binary number and you need to translate it into decimal? That's easy. Let's think about what each number represents. The rightmost digit represents the value shown, which is either 0 or 1. Moving to the left, the next digit represents two times the digit shown in that position. The next digit represents 2x2 or four times the digit shown in that position. The next is 2x2x2 the digit shown. For each digit, you are multiplying by a power of two.

In this example, we have (0x32)+(1x16)+(1x8)+(0x4)+(0x2)+(1x1), which is 16+8+1 or 25.

x32	x16	x8	x4	x2	x1
0	1	1	0	0	1

Converting from decimal to binary is almost as easy. Let's say you want to convert the number 55 into binary. At this point, it is useful to write out something like this on a piece of paper:

	32	16	8	4	2	1
55 =						

For each power of two, starting at the leftmost one, decide if the number that you are trying to convert is larger than this power of two or not. If it is, then write a 1 below the power of two and subtract that power of two from the number that you are converting. If your number is smaller than that power of two, mark a 0. Take the remaining value and determine if it is larger than the next smaller power of two and write a 1 or 0 below, depending on the outcome. Continue this process until you have nothing left of the original number. You can use this process to find the binary equivalent of any positive integer.

	32	16	8	4	2	1
55=	**1**	**1**	**0**	**1**	**1**	**1**
	55	**23**	**7**	**7**	**3**	**1**
	-32	**-16**	**-8**	**-4**	**-2**	**-1**
	23	**7**	**-1**	**3**	**1**	**0**

Decimal 55 in binary is 110111.

Addition

Math in the binary number system is actually easier to learn than it was in the decimal number system. There are fewer rules that you will need to remember. Let's start with addition. For binary addition, there are four rules that you must remember:

0	**0**	**1**	**1**
+0	**+1**	**+0**	**+1**
0	**1**	**1**	**10**

Let's look at a sample problem:

26+9=35

```
 11010
+01001
100011
```

Start by adding the rightmost digits and move towards the left, like you would when adding in decimal; 0+1=1, 1+0=1, 0+0=0. The next column presents a new problem: 1+1=10. What do you do with the extra digit? You put it into the carry position above the next column to the left, just as you would in decimal addition. This results in 1+1+0=10. Since there are no more digit columns to add, go ahead and place the 10 in the answer at the bottom.

Here is another one:

21+31=52

```
 10101
+11111
110100
```

1+1=10, 1+0+1=10, 1+1+1=Hmmm. This is not really that hard. Break this down into 1+1=10 and 10+1=11. Put a 1 in the answer and carry a 1 to the next column. The next two columns are 1+0+1=10 and 1+1+1=11.

Subtraction

Now that you have the hang of addition, let's move on to subtraction. The rules here are:

0	1	1	0
-0	-0	-1	-1
0	1	0	1 and a borrow

Like subtracting in decimal, start at the right and work to the left. In this example, we have 1-1=0, 0-0=0, 1-0=1, 0-0=0, 0-1=1, and a borrow. What you do in the case of a borrow is grab the assistance of the digits to the left to help you. For example, in the decimal subtraction problem, 13-7=? We have 3-7. You can't subtract 7 from 3 without going into negative numbers, so we use the 1 of the 13 to help us. Now we can subtract 7 from 13 to get the answer of 6. Going back to our binary problem, we can use the 1 to the left of the 0 to make 10-1=1 (2-1=1).

37-17=20

```
100101
-10001
 10100
```

Here is a problem that may help clarify borrowing for you. We start with the easy subtractions: 1-1=0 and 0-0=0. The next one, 0-1=? requires a borrow. Of course, there is no 1 to the immediate left, so we must keep borrowing until we reach a 1. The problem now becomes the one seen here:

97-5=92

```
1100001
-   101
1011100
```

011$_{10}$
1~~1000~~01
-   101
1011100

We end up borrowing from three places to the left, where we find the first 1. Now we can do 10-1=1. You will need to pad the bottom number with 0s in order to keep doing the subtraction. 1-0=1, 1-0=1, 0-0=0, and, finally, 1-0= 1.

Multiplication

Multiplication is pretty easy to do in binary. It is simply a matter of shifting and adding. In decimal multiplication, you do essentially the same thing. You need to remember a table of 100 rules for decimal multiplication. In binary, there are really two rules: 0 multiplied by either 0 or 1 is 0; 1 multiplied by 1 is 1.

```
 12
x 5
 10
+5
 60
```

Here is an example of a simple multiplication problem in decimal and binary. Looking at the example on the left, in decimal, you multiply 5 by 2 to get 10. Now you multiply 5 by 1 to get 5. You need to shift the answer one to the left, though, because you are really multiplying 5 by 10 to get 50. Now let's look at the same thing in binary:

```
  1100
 x 101
  1100
 +   0
+1100
111100
```

Notice how, for every 1 in the bottom number being multiplied, you simply copy the top number into the answer, shifted by the proper amount. The next example should make the process clearer.

214x23=4922

```
       11010110
     x    10111
       11010110
+     11010110
+    11010110
+          0
+ 11010110
1001100111010
```

See how each step in the answer either has a copy of the top number or a 0? You don't have to do all of the addition at once. Doing your addition at each step of the problem can be easier. This is what computers do when they are multiplying.

Division

```
      Quotient
Divisor | Dividend
```

You can do division in binary in much the same way that you do in decimal. It is easier to do it in binary because, once again, you don't have to know a huge multiplication table. Take your numbers and arrange them like you would in decimal long division. When you are doing each step of the division, you only need to decide if the divisor is greater than or equal to the portion of the dividend that you are working with. For example, in the problem to the left, the first chunk of the dividend is 11. This is greater than the divisor, so we can mark a 1 in the quotient and subtract the 10 from 11 to get a remainder of 1. In the event that the section of the dividend is less than the divisor, you put a 0 in the quotient and bring down the next digit of the dividend.

214/2 = 107

```
    1101011
10|11010110
   -10
     10
    -10
      010
      -10
        011
        -10
          10
         -10
           0
```

Other Types of Numbers

So far, this article has only shown positive integers. Now we'll go into how to represent numbers such as –10, 1.273, and 1.56×10^2. Before describing these methods of representing

numbers, it is important to know that, in computer systems, numbers are represented by a certain number of bits. Usually, you will find groupings of 8, 16, or 32 bits.

Negative Numbers

00010110	(22)
11101001	inverted
+ 1	
11101010	(-22)

Negative numbers are often represented using the twos complement format. To represent a negative number using the twos complement format, take the absolute value of the number that you want to be negative and invert all of its bits, making each 1 a 0 and each 0 a 1. Now add 1 to that value.

11110110	(-10)
+00010011	(19)
(1)00001001	(9)

This representation is nice because you can still do simple math with it. Let's look at the problem of −10+19=9. For this example, we will be using eight bits to represent numbers. After adding the two numbers together, we are left with the value 265! Obviously, this is not right. Due to the nature of how data is stored in a computer, if there are more bits in the answer than can be stored, the extra bits are discarded. This can sometimes be a major frustration, but, in this case, it works out to our advantage. This leaves us with the correct answer of 9. One additional thing to keep in mind about twos complement numbers is that you still can only represent as many values as the number of bits will allow. In the case of eight bits, you can only represent 256 different values. For twos complement numbers, this means that you can represent numbers from -128 to 127.

Digits Past the Decimal Place

So far, we have still only talked about integers. What happens when you want to represent numbers that are not integers? When describing integers, we would consider each digit to represent a power of two. This is true for digits to the right of the decimal point, as well.

...	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	...
...	16	8	4	2	1	.5	.25	.125	.0625	...

Let's look at 12.6875. It is represented as 1100.1011.

12.6875-8=4.6875	1
4.6875-4=.6875	1
.6875-2=NEGATIVE	0
.6875-1=NEGATIVE	0
.6875-.5=.1875	1
.1875-.25=NEGATIVE	0
.1875-.125=.0625	1
.0625-.0625=0	0

One way to represent the decimal point is to put it between two bytes. That would make representing a number with a fractional component require at least two bytes. The case shown above was a bit of a special case because the fractional component was exact. This is usually not the case with decimal binary.

Let's look at the case of representing .83 in binary. Here, we end up with .11010100. This works out to be .828125, which is pretty close to .83. If additional digits were used, the result would become closer and closer to .83. This is something to keep in mind when working with numbers that are not integers on computers or microcontrollers.

Floating-Point Numbers

Floating-point is a way of representing numbers that can have a fractional component where you don't have to rigidly adhere to a fixed number of bits to represent the digits before or after the decimal point, as was shown in the previous example. Floating-point numbers are essentially the binary version of what scientific notation is in the decimal system. For example, in scientific notation, 2.36715x10^3 represents 2367.15. The x10^3 essentially means, "shift the decimal point three places to the right." Floating-point numbers do the same thing. IEEE-754 is the reigning standard on how to represent your floating-point numbers. In this standard, there is one bit that represents the sign of the number, eight bits that represent the exponent, and 23 bits that are the number itself.

Sign	Exponent	Mantissa
one bit	eight bits	23 bits

The sign bit is a 0 if the number is positive and a 1 if it is negative. The exponent is offset by 127. In other words, if the value in the exponent section is 127, your actual exponent would be 2^0. The mantissa contains the actual digits of the binary number. Like in the decimal example, the decimal point is shifted until the number looks like this: #.####. Since this is the binary number system and the leading number will always be a 1, it will be dropped from the representation, since it is assumed to be there. You are left with what is to the right of the decimal point.

Clear as mud? Here is an example: If you want to represent -89.1875 in floating-point, you would have a 1 for the sign bit, since it is a negative number. 89.1875 translated into binary is 101101.0011. We will need to move the decimal point to the left five places to get 1.011010011. This will make our exponent be 132 (127 + 5) or 10000100.

Since the leading digit is assumed in floating-point, we can drop it, leaving us with 011010011. Add the extra 0s to this mantissa and we have completed our conversion into floating-point.

Sign	Exponent	Mantissa
1	10000100	01101001100000000000000

Floating-point has a few exceptions built into it. One

example is how you represent the value of 0. Since there is always a 1 assumed in the mantissa, this becomes a special case. To represent 0, you need to make both the exponent and mantissa all 0s. Likewise, you can also represent infinity by making your exponent equal to 255 and the mantissa equal to 0.

Useful Binary Tricks

Sometimes binary math can make things really easy for you. Remember the example of 214/2=107 that was shown earlier? If you take a look at the binary values for 214 and 107, you'll see that they are the same, except that 107 is 214 shifted one digit to the right. Shifting to the right is a quick way to divide by two in binary. It is such a useful operation that most processors have a way to shift bits around. By shifting the number multiple digits to the right, you can divide by 4, 8, 16, or any power of two. The same thing works in reverse. You can also multiply by powers of two by shifting a number's digits to the left.

When you want to find the **reciprocal** of a number, you can simply invert the bits. If we lined up all of the numbers that can be represented by the number of bits we are working with — which, in this case, is four — we get something like the illustration below. If we are looking to find the reciprocal of 2 (binary 0010), we will flip the bits so that 0s become 1s and vice versa. That leaves us with 1101, which is 13. This trick can be useful when you want to reverse the direction of travel for an actuator or if you want the speed of the drive motor on one side of your robot to be equal, but opposite, in direction.

0	1	***2***	3	4	5	6	7	8	9	10	11	12	13	14	15

0	1	2	3	4	5	6	7	8	9	10	11	12	***13***	14	15

One last trick of binary math is that, sometimes, you would like to multiply or divide by a number like 2.76, but just don't have the processor speed to do this sort of operation in floating-point math. Never fear! There is a faster way, if you can sacrifice a little precision. Find a fraction with a power of two in either the numerator or denominator that is close to the value that you are looking for. For example, if you wanted to multiply by 2.76, this is close to multiplying by the fraction 177/64 or, in other words, multiplying by 177 and then shifting the answer to the right six times to divide by 64 ($2^6 = 64$). This can give a considerable speed boost to your programs on a processor that doesn't have floating-point math built in to the chip. Make sure to always do your multiplication first or else you will be sacrificing more precision than you were expecting.

Binary math is the basis of almost everything electronic these days. Most higher level language compilers do their best to shield you from having to know about the binary number system, but, sometimes, it can be a big bonus to know what is going on behind the scenes.

Hopefully, this article has helped increase your understanding of binary numbers and their corresponding math so that your future programming will be more efficient. Also, the next time you hear, "There are 10 kinds of people in this world ..." you, too, can be in the elite club of people who aren't scratching their heads. SV

Author Bio

When not writing for *SERVO Magazine*, Jack runs Buffington Effects, a company that designs and builds animatronics and motion control devices for the entertainment industry. Check out his website at **www.BuffingtonFX.com**

New Products

ACCESSORIES

Voice Tracker™ Array Microphone

GeckoSystems, Inc. — a leading developer of mobile robot solutions — has certified the Voice Tracker™ Array Microphone for use with its CareBot™ Mobile Service Robots (MSRs). According to Martin Spencer, CEO of GeckoSystems, the combination of the Voice Tracker Array's long effective range and the robust nature of the GeckoChat's verbal interaction package enables verbal command and control and also surrogate, short-term memory assistance, such as a verbal reminder for timely medication regimens and other appointments.

As part of the certification process, GeckoSystems tested the Voice Tracker Array at ranges of up to 15 feet in environments with low noise and up to six feet in environments with high noise, while the MSR automatically self-navigated. GeckoSystems is completing the final stages for delivering an MSR platform to home healthcare (telemedicine) markets.

"We believe that our Voice Tracker Intelligent Array Microphone is useful for robotic command and control because it scans a full 180 degree field of view in front of the robot and can even pick up speech from behind," said Bob Feingold, CEO of Acoustic Magic. The Voice Tracker locates a talker and electronically steers a "listening beam" — like an acoustic searchlight — in that direction. This creates spatial filtering; sounds from other parts of the room are not picked up. In addition, constant background noise is removed by digital noise reduction algorithms. This two-stage noise reduction — coupled with increased sensitivity resulting from the continuous and constructive use of the eight microphone elements — gives the Voice Tracker outstanding range and sound quality. Mr. Spencer stated, "We are very pleased with this contemporary solution to our verbal interaction system's need for clarity for robust voice recognition."

For further information, please contact:

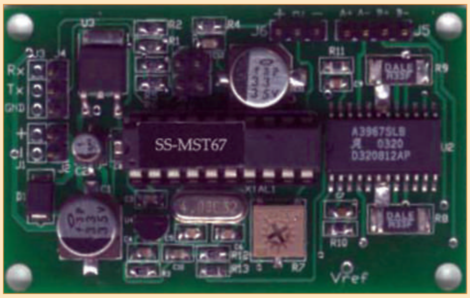
GeckoSystems, Incorporated

Tel: **678•413•9236**
Email: **MSpencer@GeckoSystems.com**
Website: **www.geckosystems.com**

Circle #82 on the Reader Service Card.

CONTROLLERS & PROCESSORS

Microstep Is the Easy Way with Super Stepper's MST67



Controlling bipolar steppers has become easier with the super stepper architecture. However, past controllers that offered only full step control were not as accurate as some applications required. That is why Avayan Electronics developed the SS-MST67 microstepping controller, which revolves around the popular super stepper architecture.

Microstepping technology enhances the motor torque to speed response, accuracy, and maximum speed. Also, slower speeds can be attained with considerably less vibration. Power consumption can be minimized, adding efficiency to battery-driven applications.

The user gets features, such as current control (up to 750 mA), speed control, position control, enabling and disabling, homing, acceleration and deceleration profiles, and more — add to that the ability to generate full, half, quad, and eight steps!

All of these features are packed on the 2.5" by 1.5" board, which can be cascaded with up to 31 other units on a single serial command link.

Don't understand how to use these boards? Avayan Electronics has devoted an entire website to acquainting users with the stepper control technology. Simply visit **www.superstepper.com** and get useful information on how to add this powerful controller to your robotic or automation project.

For further information please contact:

Avayan Electronics

P.O. Box 994
Webster, NY 14580
Email: **Avayan@avayanelectronics.com**
Website: **www.avayanelectronics.com**

Circle #98 on the Reader Service Card.

MDK-002 mARMalade ARM 720 Development Kit

EarthLCD — a wholly owned subsidiary of Earth Computer Technologies, Inc., and leading manufacturer of LCD displays, touch screen monitors, and LCD controller cards — launches the new MDK-002 mARMalade ARM 720 Development Kit, an ARM processor SBC featuring the Sharp LH79520 75 MHz 32-bit ARM 720T, assembled as a development kit with a Panasonic 7.8" flat panel color STN touch screen LCD.

Additional MDK-002 mARMalade features include 16 megabytes of SDRAM, four megabytes of compact Flash, serial/parallel/IR ports, LCD controller, 10-Base T Ethernet, and embedded Linux. mARMalade users must have an understanding of configuring embedded Linux.

Earth's mARMalade ARM-based LCD kits are available with a variety of LCD panel sizes and resolutions. mARMalade is short for Modular ARM Application LCD Appliance Device Electronics. Earth's mARMalade LCD kits meet the needs of OEMs, industrial manufacturers, and hobbyists who require a high level of integration and desire to reduce the development cycle, thereby accelerating time to market and lowering overall system costs.

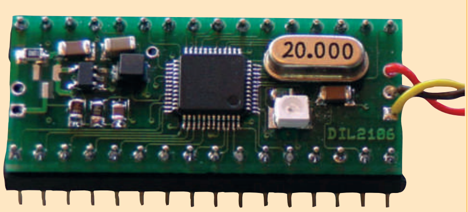
With a retail price of $299.00, the MDK-002 mARMalade is suitable for many applications, including digital visual messaging, super PDA, Internet appliance, man/machine interface, picture frame, instrumentation, home automation, or integration into a wide range of industrial applications.

For further information, please contact:

EarthLCD
Tel: **949•248•2333 x225**
Email: **tris@earthlcd.com**
Website: **www.earthlcd.com**

Circle #47 on the Reader Service Card.

DIL2106 — The New Single Board Computer



A new single board computer — DIL2106 — made by MCT Paul and Scherer is now available. Based on the widespread ARM architecture of the Philips controller LPC2106, a very compact (18 x 40 mm) and powerful design with dimensions similar to a 32-pin dual in-line EPROM was possible. The 32 pins are led to high precision contacts. It can be inserted directly into a 32-pin IC socket. An ARM version of the well-known WinECO-C serves as the programming environment. It features:

- ARM7TDMI-S (LPC2106, Philips)
- PLL-adjustable 60 MHz clock
- 128 KB FLASH-memory (ISP); 64 KB RAM
- Two UARTs; SPI, 12 C; Parallel I/O
- Two timers (32 bit); RTC; watchdog
- PWM with four channels (resolution min. 16 ns)
- Supply voltage 3.4 V to 5.5 V (typ. 15 mA at 5 V)
- Core voltage generated onboard
- Reset controller, I/O voltage regulator
- RS232 converter

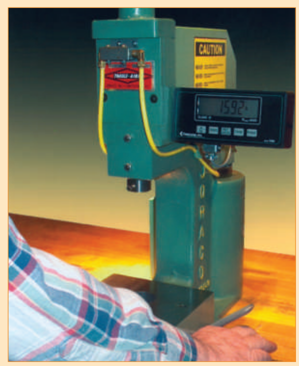
To make starting with the DIL2106 easy, MCT Paul & Scherer offers complete development kits with the DIL2106, connection cables to the PC, C compiler ECO-C-arm with IDE, hard- and software manuals on CD, and support via hotline. Call or Email for US pricing.

For further information, please contact:

MCT Paul & Scherer
Tel: **+49 (0) 38355 68740**
Fax: **+49 (0) 38355 61437**
Email: **sales@mct.net**
Website: **www.mct.de** or **www.mct.net**

Circle #117 on the Reader Service Card.

Process Monitoring Controls Monitor Presses and Rotary Indexers



A new series of process monitoring controls is available for new and retrofitted installation on Toggle-Aire® presses and assembly machines by Joraco, Inc., of Smithfield, RI.

Toggle-Aire Process Controls let users monitor ram force and position precisely and export the information to data acquisition software packages. Customers can customize their product by selecting from among a variety of display and signal conditioning devices. These features are supplied with all new Toggle-Aire presses and rotary indexing machines or they can be retrofitted.

In addition to allowing users to specify the type of process monitoring they require, Toggle-Aire Process Controls are also available for all 1/2- to 10 ton presses and rotary indexing machines. Capable of displaying ram position to ±.0005" accuracy and force to ±.05% FS, these devices can provide digital readouts on the machine and can also include RS232 or RS485 ports.

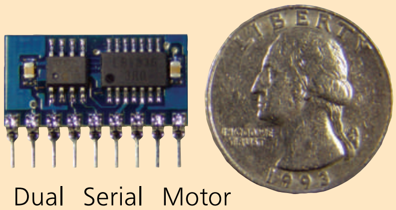
Toggle-Aire Process Controls are priced from $2,495.00, depending upon configuration. Literature and quotations are available upon request.

For further information, please contact:

Toggle-Aire® Division of Joraco, inc.
347 Farnum Pike
Smithfield, RI 02917
Tel: **888•889•4287** Fax: **401•232•1711**
Email: **joraco@worldnet.att.net**
Website: **www.joraco.com**

Circle #76 on the Reader Service Card.

Pololu Micro Dual Serial Motor Controller



Pololu announces the availability of its latest motor controller, a new version of the popular Pololu Micro Dual Serial Motor Controller. The nine-pin module contains two independent H-bridges and all the necessary control logic to simplify motor control. The new model maintains the same tiny package as the original, but incorporates many new features requested by robotics enthusiasts. The motor controller supports independent, bidirectional control of two DC motors with currents of up to 1 A per motor. The motor supply voltage can be as low as 2 V, making the controller perfect for small toy motors. Multiple units can be daisy-chained on one serial line to control up to 62 independent motors. With automatic baud rate detection and a simple protocol, it is easy to add motors to any microcontroller project.

The latest motor controller incorporates three exciting new features. The user can now reprogram the motor ID numbers so that different part numbers do not need to be ordered if two or more modules are used on one serial line. A new operation mode allows the two H-bridges to be used in parallel so that one motor controller can provide a single motor with up to 2 A. Electronic braking is also available, allowing an even greater degree of control. The price for one unit is $23.00.

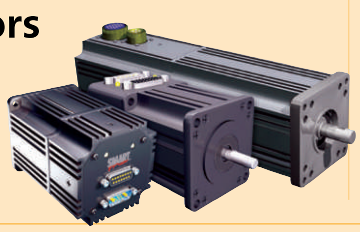
For further information, please contact:

Pololu Corporation
6000 S. Eastern Ave., Ste. 5-E
Las Vegas, NV 89119
Tel: **877•7•POLOLU** or **702•262•6648**
Fax: **702•262•6894**
Email: **www@pololu.com**
Website: **www.pololu.com**

Circle #140 on the Reader Service Card.

MOTORS

Integrated Motors Benefit the Industry



Animatics Corporation offers high performance, integrated servo systems in its SmartMotor™ product line. Each SmartMotor is an all-in-one, integrated servo motor system that combines a closed-loop servo motor, amplifier, encoder, and controller to optimize machine control and automated tasks.

Compared to the technology of typical servo systems that historically consist of individual controls, drive amplifiers, and motors, this all-in-one design is revolutionary.

With capabilities of multi-axis coordinated motion and G-Code software, CNC motion becomes easier to operate. By utilizing highly flexible onboard I/O and additional expanded I/O, any SmartMotor can control an entire machine. Frame sizes range from standard NEMA 17 to NEMA 56 with speeds up to 18,000 RPM.

The simple design of the SmartMotors has brought these important benefits:

- Much smaller control cabinet. The cabinet can also be eliminated entirely (no drive amps or controls to install).
- Less cabling (no command signal, encoder feedback, or motor amp).
- Easily controlled by a simple PC or HMI (Human Machine Interface) or programmable in Basic to stand alone.
- Fewer parts means longer life and less downtime.

Standard features that come with the SmartMotors — depending on the size chosen — include:

- On-the-fly variable electronic gearing.
- Efficient, battery-driven operation.
- Electronic camming with relative and absolute cam modes.
- True 3-D coordinated motion (not limited to any plane)
- RMS voltage and current feedback for true torque sensing.
- Both RS-232 and RS-485 serial communications operating simultaneously.
- Data mode communications that allow for specific communication protocols.
- Up to 64 channels of expanded I/O per motor via the Anilink protocol.

Optional devices include:

- Built-in automatic brake powered by the SmartMotor.
- High speed I/O trigger option.
- IP65 protocol with environmentally sealed connectors and joints.
- Precision straight and right angle planetary gear heads.
- Isolated communications and power DUN rail mount distribution block.
- Isolated DIN rail mount motor I/O break-out that accepts standard optical modules.

SmartMotors have been introduced into nearly every type of manufacturing process including: animatronics, semiconductor handling equipment, medical, automotive, CNC equipment, and countless other manufacturing sectors.

Some of the uncommon applications integrated motors are used for include: autonomous and remote controlled robotics, human head impact studies, and robotic undersea devices.

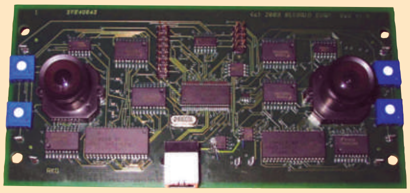
For further information, please contact:

Animatics Corporation
3050 Tasman Dr.
Santa Clara, CA 95054
Tel: **408•748•8721 x332** Fax: **408•748•8725**
Email: **jlaznicka@animatics.com**
Website: **www.animatics.com**

Circle #132 on the Reader Service Card.

ROBOT KITS

THEIA Integrated Camera System



The new THEIA Integrated Camera System is a 1/2 inch, full-frame CMOS imager that provides high quality, precisely synchronized stereo image capture. Using micron high speed cameras, THEIA is an image system with variable capture rates, not a video system. It comes standard with programmable shutter speed, onboard frame storage, and a USB 2.0 port. Additionally, THEIA uses a single cable for data, power, and control; it also allows multi-camera synchronization and progressive scanning. THEIA software runs on Mac or Linux and allows you to focus the lenses, calibrate the camera, and get basic range data. SEEGRID also offers free, one-day recalibration service for THEIA systems.

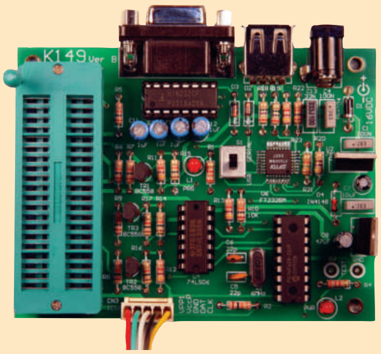
For further information, please contact:

SEEGRID Corporation
Tel: **412•731•7174 x115**
Website: **www.seegrid.com**

Circle #38 on the Reader Service Card.

TOOLS & TEST EQUIPMENT

USB/Serial PIC Programmer



USB ports are almost standard today on all PCs. This PIC programmer uses the USB port, but, just in case, it also has a standard serial port connection.

Features:
- USB/serial port connection
- Software that can be updated for new PICs
- Windows 9x/NT/2000/XP compatible software
- ICSP (In Circuit Serial Programming) connector
- Quality PCB and silkscreen for easy assembly
- Complete assembly and programming instructions

Specifications:
- L: 4" W: 3" H: 5/8"
- Requires external 16 volt DC @ 50 mA power supply
- Requires USB or serial cable

The MicroPro Software supports and will program a variety of PICs. You can add new software on your own and new updates are released about once a month, so new PICs will be supported.

For further information, please contact:

Carl's Electronics, Inc.
17 Laurelwood Rd.
Sterling, MA 01564
Tel: **978•422•5142** Fax: **978•422•8574**
Email: **sales@electronickits.com**
Website: **www.ElectronicKits.com**

Circle #67 on the Reader Service Card.

MC-7 (Meter Calibrator–7)

Precision Resistor Company is introducing the MC-7, a packaged group of seven selected resistor values for use in the verification and adjustment of the resistance function of digital multimeters. The seven resistance values of the resistors are 1, 10, 100, 1K, 10K, 100K, and 1M Ω with a rated tolerance of .01% for values above 10 Ω and .02% for the 1 and 10 Ω units. The resistors are contained in a plastic case with the leads extending through the sides. At $63.90, this resistor selection offers an economical solution to the task of calibrating the resistance function of 3-1/2 and 4-1/2 digit multimeters.

For further information, please contact:

Precision Resistor Company
10601 75th St. N.
Largo, FL 33777
Tel: **727•541•5771** Fax: **727•546•9515**
Email: **prc@precisionresistor.com**
Website: **www.precisionresistor.com**

Circle #104 on the Reader Service Card.

Promotional Robots

by Tom Carroll

In my last article, I talked a bit about Westinghouse's Electro — the Moto-Man — Servant of the Future. Needless to say, the only people Electro ever served were Westinghouse's marketing team. His only responsibility in life was to capture the attention of the potential buyers of Westinghouse's products so that the marketing people could do the final sale. He was really the first promotional or show robot, even though he was actually controlled by a mobile teleoperator.

For that matter, what is a teleoperator? If you look at the word, it means performing operations from a distance or doing some sort of work by remote control. You've seen pictures and educational TV programs that have shown people using remotely controlled robot arms in nuclear hot cells where an operator sits safely outside the cell and looks in through radiation-opaque lead glass at radioactive objects. His hands manipulate two robot sending hands from outside the cell as the remote hands mirror the movements inside the cell to grasp and move items. These first devices were purely mechanical and their movements were accomplished via moving cables, rotating and sliding shafts, and moving chains on sprockets.

Later devices emerged that accomplished motion through a series of small encoders on the sending hands, in addition to electric servo motors and encoder feedback devices on the receiving hands. These electronic techniques allowed for much greater distances between the operator and teleoperator sites, especially with the use of television links to provide the visual feedback. Doctors have actually operated on patients from across oceans using satellite links for the two-way control signals and the stereoscopic video vision.

Despite the cutting edge technology here, these devices are still considered teleoperators. The doctor/operator must still be in the loop. She can't tell a robot, "I'm going out for a cup of coffee — finish this operation for me." Likewise, an Army sergeant can't tell a remotely piloted vehicle (RPV), "I'm tired. I'm going to take a nap. If you see anyone who looks like Osama bin Laden, blast him with a bomb. Wake me when it's over."

We're getting close to that, but we're not quite there yet. Yes, the sophisticated Predator RPV can be programmed to take off, make a long mission, take a series of intelligence photos, return to base, and land — all automatically. We still have the need for a person in the loop for target identification and elimination or, in the case of the surgical robot, nation and delicate surgical procedu the person — through

the person — through visual feedback — determines how to control the robot to cause the necessary movements. RPVs can even deliver ordinance, but the target needs to be verified by a human and bomb damage assessment must be performed afterwards.

A Little Background

You're probably wondering just what a promotional robot is and how it differs from a teleoperator. Aren't promotional robots just over-sized, radio-controlled toys? The answer to that last question is probably no, unless you consider a machine that costs between $20,000.00 and $70,000.00 to build a toy. Yes, the intelligence and sensor suite are not microcontroller/software packages connected to a series of ultrasonic transducers and LED/phototransistor modules situated around the periphery of a typical experimenter's robot.

The intelligence is the operator's brain, his eyes are the sensors, and his past experience is the software. These robots still have the same mechanical requirements; most are far more mechanically complex than the typical experimenter's microcontroller controlled mobile robot.

Some of these promotional robots are pretty unique, but it's not the cost of the machines or the construction techniques used that make these special breeds of robots so interesting — it's the backgrounds of some of the early people involved. If you think opposing lawyers in a high profile trial act like crooks and mortal enemies to each other, just allow me to dig into the early movers and shakers of the promotional robot business. Some of the cloak and dagger antics of these people remind me of James Bond movies.

If one developer found a particular supplier that really worked well for his design, he would paint over or scratch off the nametags just to prevent anyone else from knowing where the parts came from. They would use P.O. Box numbers on their business cards so no one could find their secret shops, which were, in many cases, just their own garages.

One trick that many of these first developers were good at was borrowing another popular style of robot to copy it. "Just for this one show — I'll bring it back in a few days," they would say. This person would then remove the fiberglass shell pieces and make molds from the shell parts and return the re-assembled robot to its owner. They could now produce all of the robot shells and robots that they wanted to.

Most of the idea people had no mechanical or electrical talents — they were just very good at the PR game. They would hunt down people who they had heard about to do the actual design and construction. Many of these true robot designers were promised ground floor opportunities in an exploding new business when, in fact, the PR people just picked their brains and had the robots built. The PR idea person now had a valuable robot that could make him over a $1,000.00 a day. He hired operators for perhaps $100.00 a day — not a bad profit margin.

After attending a technical show back in the mid 80s, I happened to catch the operator of a promotional robot out behind the convention center. It turned out that he was the owner of the robot and not the normal operator, whom he had just fired the week before.

I had noticed a large plywood board over the floor in his van and that started our conversation. It seems that the operator had finished a show and quickly loaded the robot into the van without adequately securing it with bungee cords. He steered around a sharp corner and the robot tipped over. The operator quickly stopped, picked up the robot, tied it securely to the wall of the van, and continued on to his house. He parked in his garage and went into the house for the evening. It was a Friday night and he did not go to his van all weekend.

TOM HITCHES A RIDE

On Monday morning, he went into the garage and smelled something funny. It seems that the car-type battery inside the robot had tipped over and had stayed tipped over — even after the robot had been set upright. The sulfuric acid from the battery had all drained out onto the robot's electronics, the aluminum/steel/wood base, and the two drive motors — ruining them all. (The owner showed me the corroded robot base in a box in the van.) To top that off, it ate a hole in the floor of the van and even damaged the garage floor.

The robot's owner scrambled around to try to find replacement parts. He hadn't built it — it was probably a knock-off of someone else's robot — and so finding the parts was difficult. He managed to find a person whom he called a computer whiz — a local 16-year-old high school kid who rebuilt the robot for him.

I was amazed at how he used

model aircraft servos to drive pots that, in turn, drove what I believe was a PWM circuit that was, in turn, driving power Darlington transistors and relay polarity-reversing. It worked! He still was using a liquid electrolyte car battery, but he tied it down this time and put it in a plastic boat battery case. Maybe after a second robot destruction by the battery, he might have started using gelled electrolyte batteries. I'll bet he wished that he had known about Vantec's RET-4 electronic throttles back then.

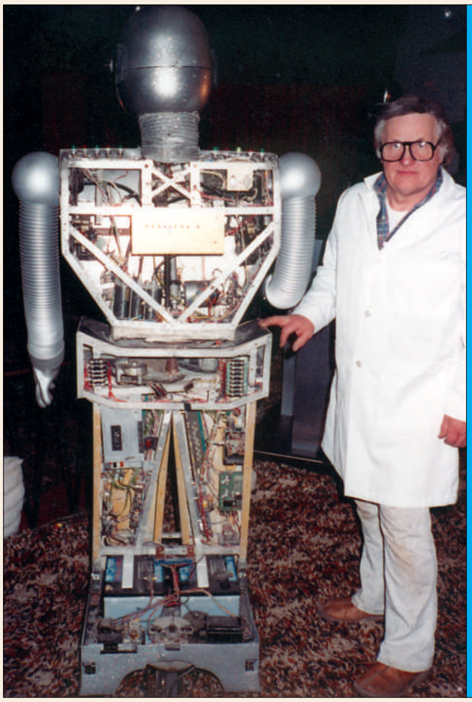
The Robots

There have been many robots built by people over the years, but one robot in particular truly made a big splash in the news back in 1975. The builder and his house were every bit as extraordinary as the robot. David Geer featured this unique robot — named Arok, built by a fellow who lives near Chicago, IL — in the January 2004 issue of *SERVO*. Ben Skora (note his name and his robot's name) did not intend for the robot to be a promotional robot; it was just the culmination of one of his life-long dreams. Arok was featured in many popular technical magazines almost from the time of its conception.

I was lucky enough to be able to visit with Ben in his amazing house during some off time while I was at one of the RIA/RI-SME (Robot Institute of America/Robotics International — Society of Manufacturing Engineers) robot conferences in Chicago in 1983.

After a tour of his very unique house, Ben allowed me to get into the insides of his robot and take some photographs. Arok was giving me a ride in Figure 1 and Figure 2 shows his pre-microcontroller/H bridge innards. Using automotive motor assemblies, other surplus items from the 70s, and telephone DTMF techniques, Ben was able to control Arok and store the control signals as programs.

Arok had such powerful arm (shoulder) drive motors that he could grasp the handle of an upright vacuum cleaner with his hand and strong-arm it out in front of him with the motor part a full six feet in front of his body. I don't know any humans who can do that. Arok was well-built by anyone's standards. Reread David's article from January to see just how advanced Ben and Arok were for their time.


AROK AND HIS CREATOR

The First Promotional Robots

The promotional robot business — or show robot business, as it was sometimes called — actually started a bit before Arok was perfected by Ben Skora. In the late 60s, a New Jersey entrepreneur by the name of Tony Reichelt formed a company named Quasar Industries (not to be confused with the proto-typing shop in Rochester Hills, MI). Reichelt developed a series of promotional robots that were fairly easy to produce: a spherical head atop a conical body. There were some futuristic shoulder ornaments to make the robot look space-inspired (Figure 3).

Reichelt called his design Klatu (u-talk spelled backwards) after the robot star of the 50s movie *The Day the Earth Stood Still*. Someone once told me that the Klatus reminded him of ice cream cones upside down with the ice cream placed on the wrong end, but, jokes aside, that design was quite popular for many years. Reichelt and Quasar probably built more promotional robots than anyone else.

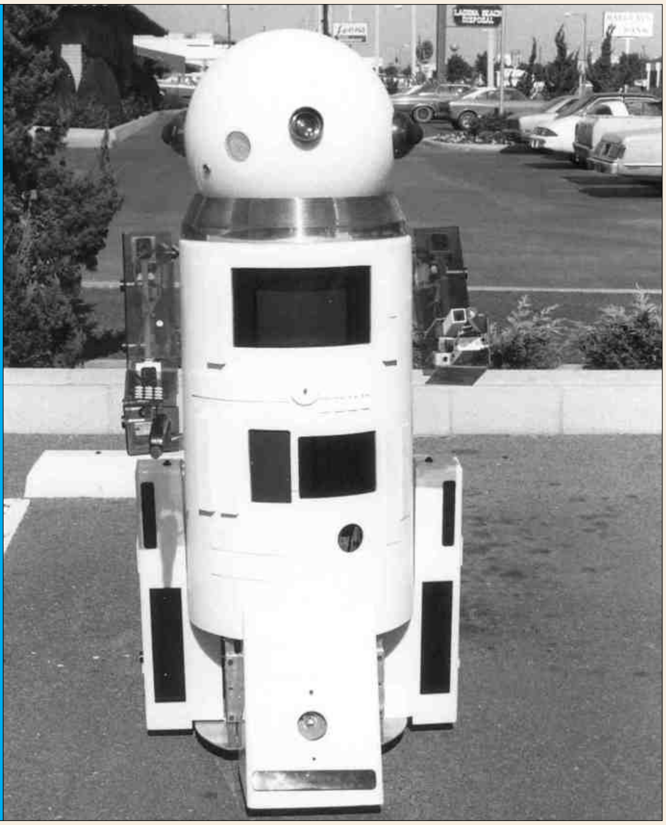
Unfortunately, Quasar generated more negative publicity than positive. News articles with the following sentiments began popping up: "A private American company, Quasar Industries, launches a robot selling for $4,000.00, which is supposedly able to do anything a domestic maid could do — and more — including teaching the kids French." In 1977, the *Vancouver Sun* reported on a domestic android manufactured by Quasar Industries that could, "serve your dinner, vacuum your rugs, baby-sit your kids, and insult your enemies."

According to Quasar, the robot stood five feet, four inches tall, weighed 240 pounds, and was called the Domestic Android Robot. Quasar promoted it as programmable and said it could, "perform a dozen basic household tasks, such as mopping the floor, mowing the lawn, washing dishes, and serving cocktails." It came equipped with, "a personality and speech," so that it could, "interact in any human situation." It could, "teach the kids French," and, "continue teaching them while they sleep." At a price of $4,000.00, the machine seemed a real bargain.

Phil Karlton — a researcher at Carnegie-Mellon University — was heavily involved in exploring artificial intelligence, speech recognition, and related research problems, so he knew a thing or two about robots. News items about the Quasar robot

FUTURISTIC KLATU

began appearing on his site on the Arpanet — the government's early version of the Internet. All of the researchers knew the thing was a hoax. Someone on the net mentioned a rumor that was on the radio claiming the Dade County, FL police department was considering purchasing a Quasar guard robot for the county jail for $7,000.00.



ENERGY SPOKESBOT EFFIE

One of Quasar's domestic robots — Sam Strugglegear — was situated at a large department store in downtown Pittsburgh, PA. CMU AI researchers were sent to investigate. They reported: "In the men's department, among the three-piece suits, was a five-foot-four 'aerosol can on wheels,' talking animatedly to a crowd. Electric motors and a system of gears moved the device's arms. The robot seemed conversant on any subject, recognized the physical features of customers, and moved freely in any direction. The crowd was charmed."

These researchers knew better. They looked around for evidence of someone controlling the robot. "Sure enough," they said, "about 10 feet from the robot, standing in the crowd, we found a man in a blue suit with his hand held to his mouth, certainly hiding a wireless microphone. We watched for awhile and noticed that, whenever the robot was talking, so was the man in the blue suit — muttering into his hand. The man had a wire dangling suspiciously from his waist."

Yes, I had read the hype about Quasar's various types of robots and managed to see a few of them while visiting different cities on business trips. It was a simple robot with one degree of freedom arms — certainly not capable of any sort of domestic work around a house. Knowing this, I was nonetheless entertained every time I saw one of these robots. After a bit, I could always spot the operator and sometimes teased him a bit by engaging him in conversation when the robot was supposed to be "talking" with someone.

It was always cute to see little kids enthralled by these huge creatures from the movies and TV, but it was the adults' reactions that I enjoyed most. Grown men and women would regress to grade school status as they "talked" with the robots. Most people were fooled and that's all the sponsors and Quasar wanted. I still find it hard to believe that Quasar thought they could get away with their claims after someone actually bought one and found out that it could not perform as advertised.

With all of this publicity over relatively simple machines, it was natural that others wanted a piece of the action. Gene Beley — a Southern California businessman — had an amusement arcade game business when he heard of Quasar; he was one of many company representatives. Gene was and is an honest businessman who just wanted to develop a business that he could enjoy.

After being stung by some of the claims that came back to bite him, he met Ray Raymond — who had been in the restaurant furniture business — and they collaborated to build their own promotional robot. They built a robot that was tentatively named Effie Conserva as a spokes-robot for an energy conference (Figure 4).

Gene then decided to branch out on his own and called his new company Android Amusement. He developed several models in a series of robots — the DC-1 (Drink Caddy) robots (Figure 5), the DC-2, and Mr. (T) Telebot (Figure 6). Notice the triangular base similarities in the two robots pictured. The first of these DC-series robots was made out of a blue, blow-molded, plastic chemical drum. Beley also developed two other robots that he called Adam and Andrea Android, which were animated mannequins atop motorized carts.

At first, he had a hard time finding just the right components for his robots, as there weren't too many robot supply companies at the time. The word was spreading that BEC — the British wheelchair company — made some great wheel/motor assemblies that worked great for robots and Gene made use of these for his robots. A Torrance, CA company with a similar name — A-BEC Electric Wheelchairs (who later moved to Fresno as Sunrise Medical) — stocked these rather expensive ($500.00 each), but very tough, motorized wheels.

The rest of the robot consisted of a pair of gelled electrolyte batteries and a model aircraft R/C system with the receiver in the robot driving sets of Vantec or Futaba single or dual electronic throttles. The arms, head, and other special effects were controlled by the other R/C channels. A wireless microphone system and an amplifier connected to the receiver served as the robot's "speech."

In the beginning, the designers and builders of these robots used R/C systems operating on 72 MHz — systems designed specifically for model aircraft use and not ground use, such as model cars and robots require. The reason robot builders used these systems was because aircraft systems offered seven to eight channels — not the two or three channels available in car 75 MHz systems. In response, Dail DeVilleneuve of Vantec and a few others have been offering the seven to eight channel re-tuned systems in the 75 MHz band that more recent robot operators have been using.

One of the problems with the typical promotional robot setup was the presence of the operator, who was easy to spot as he tried to manage controlling the robot's operations while simultaneously speaking into a wireless microphone. Most would use a pre-packaged Futaba or similar R/C system transmitter powered by a cordless drill battery pack for long operation time, which was strapped to the operator's back under a coat. A multi-conductor cable ran up their sleeve to a small, hand-held control box with a tiny joystick for wheel control and several pots for other functions. The operator would mumble into his other hand, which hid a wireless mike for the robot's "voice." One operator cleverly used an un-lit pipe to hide a mike hidden in the mouthpiece.

I helped one operator hide his controls by building them into a gutted 35 mm camera body. A tiny, two-axis joystick on the top controlled the robot's two differentially-driven wheels and three slide pots on the back controlled the head and arm movements. To have the robot "speak," he raised the camera to his face — as if he was taking a picture — and spoke into a small mike in the lower left hand corner of the camera's back. I later added eight buttons on the camera body that allowed

NORM! — NOT QUITE

BEFORE MOBILE DVDs

the operator to key up one of eight pre-recorded, 20 second messages from eight digital voice recording chips; these chips were triggered by a Vantec Keykoder inside the robot's body. (Go to **www.vantec.com** for more information.)

In the mid 80s, it seemed like everyone was trying to get into the business of building and operating promotional robots. In the beginning, the early operators could get $1,200.00-$1,500.00 a day, plus expenses, for operating these machines. Some companies — like The Robot Factory in Colorado Springs, CO — have been in business since 1966 and are still going strong. These are the people who still offer good products without all of the marketing distortions. Gene had quite a run with his series of robots, but decided to get out of the business and is now the editor and publisher of a newspaper, *Country News*, in Morgan Hill, CA. His college education was in journalism and his hobby is now collecting Jaguars (the cars — not the cats).

There is still room for honest people who enjoy entertaining the public, have a talent for electrical and mechanical things, and want to build and operate promotional robots. There is a lot of information available on how to build a large robot and remote control system. Books, such as *Build Your Own Combat Robot* — which Pete Miles and I co-wrote several years ago — are a good start. There are even some newer books that deal with mechanical systems, motors, batteries, and R/C systems that you will find listed here in *SERVO* or in *Nuts & Volts*.

Remember, when referring to combat robot information, you are dealing with another breed of machine. Combat robots are designed for short bouts of intense energy. Motors are fed hundreds of amps and components are stretched to the limit. Structures are much more robust in order to take weapons beatings. On the contrary, everything you can imagine has been turned into a promotional robot — from a Coke can, a robot banana, fuzzy creatures riding bicycles, to virtually anything conceivable.

Promotional robots operate at a much lower intensity; however, they must be very reliable over long stretches of time — like an eight-hour day. They must be easily reparable in the field by operators who may not know as much about robots as you do.

Action prop R/C robots for movie studios are another similar category that falls between the two. Props are built to last for only a short time, but can take a beating by the studio special effects people, grips, and electricians.

Building large robots is very rewarding. New microcontroller systems, high power H bridges, better motors and batteries, higher frequency wireless microphone systems, and better materials can result in a robot that is vastly superior to the first machines that I mentioned. If you're tired of watching your small platform snake its way about your kitchen floor — only to be tackled by your cat — try your hand at a large-sized robot. At least your neighbors and their kids will be amazed. **SV**

Send us a high-res picture of your robot with a few descriptive sentences and we'll make you famous. Well, mostly. menagerie@servomagazine.com

LegWay

Steve Hassenplug, Lafayette, IN

This is a two-wheeled, self-balancing robot made entirely from LEGO bricks. Instead of an accelerometer, it uses two EOPDs (electro-optical proximity detectors) to measure the distance to the floor.

To move forward (for line following), LegWay actually sets the motors to run backward, causing a tilt — which it automaticly corrects by moving forward. When one sensor is over the line, it stops that motor and balances using only the other motor, causing it to turn.

To spin in place, both motors are shifted "off center" in opposite directions — the same amount — but they still correct for tilting.Processing is done in a standard RCX brick, but the code was written in the native BrickOS (LegOS) to get the servo loop frequency up to 20 Hz. Videos and full build instructions are available on my website.

www.teamhassenplug.org

Meccano Man

Dan O'Connell, Canada

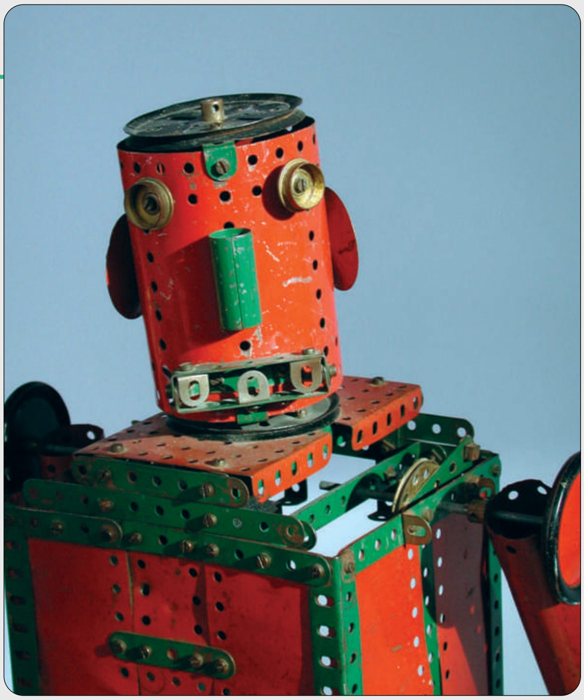
Well, I suppose he's not really a robot, but he sure seemed like one when I built him at the age of 12. The old Meccano company in Britain is long gone, but, as a kid, Meccano was *the* way to build something mechanical.

After learning a little about metal working, I can see where corners were cut, screws came loose, or the metal was too thin. It was next to impossible to make something that stood up.

The motor was particularity irksome. I remember having to save my pennies to buy it, but now, when I look at it, I'm pretty disappointed with the quality: no bearings — not even bushings, a really sloppy fit to boot, and — for some reason — it needs 20 volts.

In spite of such deficiencies, Meccano is making something of a comeback. Adult builders have formed clubs in many countries and yet another company has started to produce "Meccano" parts.

www.robot-one.ca



GEER HEAD

by David Geer
geercom@alltel.net

Medieval Automatons

Have you seen those devilish little demons — Pinhead, Tunneler, Jester, and Blade — from the B-movie The Puppet Master *(and its sequels)? If so, you've had a glimpse of crudely manufactured life forms that are comparable to eerie, ornamental, Medieval automatons!*

Pre-Medieval History

The philosopher and teacher Aristotle was the first man on record to propose the idea of automation — machines working on their own.

He not only suggested automata (the correct plural form of automaton), but also stated that, if we could build them some day, we wouldn't need laborers for simple or repetitive tasks.

In various times between the period when Aristotle wrote and when robotics became a reality, we have gotten glimpses of the mention, appearance, and mastery of history's animated automatons.

Mechanical soldier, Boilerplate, in the lime light (19th Century). Photo courtesy of Paul Guinan, **www.bigredhair.com/robots**

We're focusing here on automata that arose in the Medieval period. However, there are records of their creation or design long before 500 AD and certainly many examples exist today.

Don't believe me? Have you ever seen ballerinas that dance in music boxes, jack-in-the-boxes, wind-up monkeys with little cymbals, fortune teller machines,

Title Picture — Inside a bird automaton. Image courtesy of Ray Bates, **www.thebritishclockmaker.com**

and other moving figures that greet or entice us at arcades and carnivals? How about animatronics at Disney World?

Now, When Was That?

Few sources put an exclamation point on when the Medieval period began or ended. The most common figure is from 500 AD to 1450 AD (though some reports give us all the way into to the 1500s, allowing us to include Leonardo Da Vinci).

Much Good From Medieval

Though notable automatons were developed in Asia, Spain, and elsewhere in the world, we'll be talking primarily about Europe. The history of the Dark Ages in Europe — also called the Middle Ages and the Medieval period — somewhat paralleled the outline of the early history of America.

People from various nations came and settled throughout Europe. Wars from without and within, poverty, disease, and famine were the incubators for a very young continent.

From this period, we have inherited history, weaponry, hierarchical social structure, architecture, trade, surnames, tournaments, and automatons.

Automatic

Automata were early attempts at creating creatures like animals, plants, or humans by means other than nature simply taking its course. Absent genetics and cloning, automata were made from wood, ceramics, cloth, and other materials that could be fashioned, rather than being born out of flesh.

Later, automata were run by clockworks and then even by steam and electricity.

Though not always useful, Medieval automata were entertaining and fascinating to the people of the time, being invented only by the most genius of craftsmen.

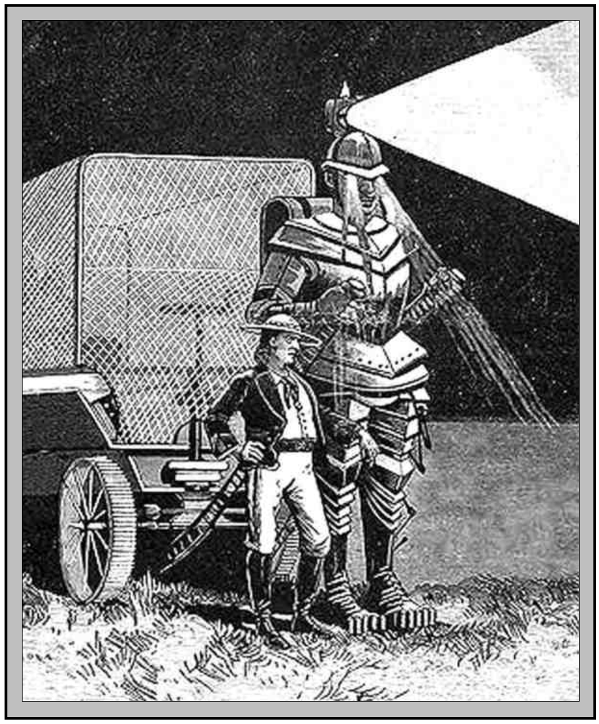
Automata Craftsmen

Leonardo Da Vinci (1450-1500s — start of the Renaissance) and Al-Jazari (1300s) were among these great geniuses of automata. Da Vinci's creations included the iron man and the automatic calculator (the first true computer, 150 years ahead of Pascal's).

Al-Jazari hatched his peacock automaton — among other creations — which kept the people spellbound. These were mechanical creations that few, if any, could duplicate.

Al-Jazari wrote a book on automata/automation that is still searched and sourced by engineers today.

As another example of an automaton, the book contained an illustration of an automated woman who filled a wash bowl and emptied it again. During his career, Al-Jazari held the status and position of what might be called Engineer to the King.



This giant electronic man of the 19th century obviously pulled his own weight.
Photo courtesy of Paul Guinan.

Clockmakers

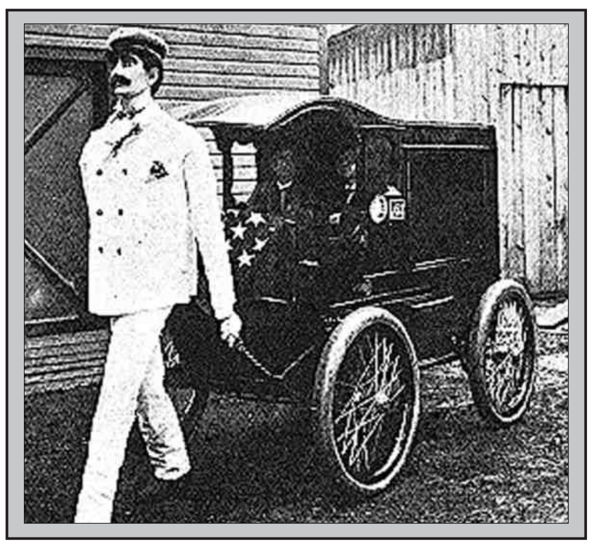
Clockmakers were frequently automaton makers, as well. Predating today's cuckoo clocks, these engineering creations were clocks with automata or even automata based on clockworks — without the clocks!

Automata of the day adorned churches, clock towers, and smaller household clocks; some of these amazing creations were strictly ornaments to be used only for entertainment.

There was an explosion of automaton crafting in the 13th century as these clockmakers and their amazing creations came into great popularity.

These automata can best be described as automatic in movement through a complex series of clockwork gears and axles.

Where he leads, you will follow —
especially if you're in the carriage (19th Century).
Photo courtesy of Paul Guinan.



The top dial on the Astronomical Clock in Prague.
Image copyright 2002 by Stephen and Mikel Milton.

Astronomical!

One example of a noted clockwork automaton is the giant Astronomical Clock — constructed in the 1300s — still working and keeping perfect time in its home city of Prague in the Czech Republic.

When the clock — which is set in a tower — chimes the hour, the entertainment begins. A figure in traditional Turkish attire draws a sword as a skeleton rings a bell. Then a door opens and several automaton Apostle figures move across the top of the clock.

The residents of Prague were so jealous over their new clock that they put out the clock-maker's eyes so that he could never see to build another one elsewhere. They wanted theirs to be the only one in existence.

A short time later, the clock was in need of repair, but no one was available to fix it. It was some time before it was put in working order again, but, once it was repaired, it ran for centuries after.

Tourists flock to see the Astronomical Clock — one of the few pieces of remaining evidence of this type of engineering and craftsmanship from the 1300s.

The clock displays not only the time, but also the revolving of the stars, sun, and moon.

Three gears of various sizes on one axle — each with various numbers of teeth — separately rotate the moon, sun, and stars. On the hour, the skeleton, Turk, and Apostles appear. The skeleton's bones even rattle when he strikes the bell!

The clock has been well-maintained, undergoing several repairs and updates. A calendar dial was added around 1500. The clock was also decorated with stone scriptures and, later, moving statues were added to the side of the dial.

Post WW II, the clock underwent a complete renewal, having been damaged in the war. A community effort restored the clock. At some point, the clock was reconfigured to show Old Czech Time, Central European Time, and Babylonian Time, as well.

Automata — the Real Goths

Gothic clock tower automata would present themselves as moving figures that came out of doors, struck anvils to sound the hour, and then disappeared through another door. It was in Orson Welles' classic film noir, *The Stranger*, that an automaton just like this killed Welles' character just after he had finished repairing the antiquated robot.

In the movie, a war crimes investigator is tracking down Welles' character — Franz Kindler, leader of the Holocaust.

As the plot goes, he had gone deep undercover, having erased every trace of his true identity. Kindler's wartime colleague is released from prison to lead the investigator straight to his prey.

Kindler kills the former comrade, leaving the investigator, Wilson, with only one remaining clue – Kindler's keen interest in antique clocks, the very kind that house automata in need of his care.

It's in a closing scene in a church steeple in Harper, CT where the automaton, not Wilson, does Kindler in for his crimes.

Other examples of clock-based, clockwork automata include the rooster seated on a cathedral in Strasbourg, Germany (1350) that crows and flaps its feathered wings precisely at noon each day.

Another example is the two gigantic automata used to sound a bell above the great clock tower in Piazza San Marco, Venice (1497).

Unfortunately for their makers and us, the names of these craftsmen were not preserved in history; their great masterworks were simply ornamental in nature. We are actually fortunate to have what records we do of the creations themselves.

Sources

Paul Guinan
www.bigredhair.com
Guinan@bigredhair.com

Michael Ravnitzky
*Acknowledged for his research contribution
Mikerav@mindspring.com

Cornelis Robat
Editor in Chief
The History of Computing Project
www.thocp.net
editor@thocp.net

Stephen and Mikel Milton
www.milton.com/praha2002
milton@isomedia.com

As time progressed — beyond the Medieval period — craftsmen became known, recorded, and praised for specific works. These include Baron Wolfgang Von Kempelen's chess playing Turk (1769) and Jacques de Vaucanson's Mechanical Duck (1738). Other, later automata include Droz's automatic scribe (1774), which wrote messages of up to 40 characters.

Specific Example of Automaton Activity, Popularity, Mystery

As with today, once an automaton engineer solved a problem that would facilitate a new automaton activity, other engineers followed suit. There were several iterations of chess playing automata, for example. They were creations of those who were sometimes referred to as technician magicians; in fact, automaton chess players were often shown as magic tricks.

The most famous was the chess playing Turk of Bratislava, first exhibited around 1770 in Vienna, Austria. A human chess player operated the Turk for each move. Obviously, it did well when operated by a champion, such as William Lewis.

The Turk was a mannequin, clothed in a turban and cloak. He sat in front of a maple cabinet that was 4' x 2' x 3' and was mounted on a wheeled base. The Turk went from owner to owner performing around the world and the truth about its operation wasn't revealed until almost a century later.

Automaton Never More?

Near the beginning of the last century, engineers instilled automata with electromechanical technologies, helping them evolve into what were soon coined and are now referred to as robots.

In little-known parts of the world, you can still find the more antiquated automata well preserved for viewing and entertainment purposes and, of course, for a fee.

As we've alluded to, there are automata, or automatons, pre-dating 400 BC, as well as those created up to the present time. See the resource list for more information and check 'em out! SV

Jack the Smiter®, built around 1480, strikes time on his bell.
Image courtesy of Ray Bates.

Resources

1. Graham Cluely has some interesting images of the Astronomical Clock:
www.grahamcluley.com

2. The Miltons provided an image of the Astronomical Clock for this article:
www.milton.com

3. Automation Hall of Fame:
www.automationhalloffame.org/timeline_01.html

4. About Leonardo Da Vinci:
www.mos.org/sln/Leonardo/InventorsWorkshop.html

5. Home of some automaton images:
www.thebritishclockmaker.com/automata

6. Automata Information: **www.marcdatabase.com/~lemur/dm-delightful-machines.html#automata**

7. Gallery of Automata:
www.nyu.edu/pages/linguistics/courses/v610051/gelmanr/ling.html

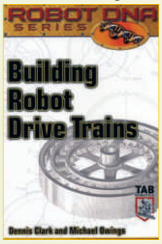
8. Robots and Automata:
www.bigredhair.com

9. Automaton Theatre: **www.teatrodeautomatas.com/home_eng.html**

Mind Candy For Today's Roboticist

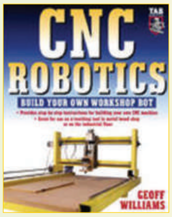
The *SERVO* Bookstore

Building Robot Drive Trains
by Dennis Clark / Michael Owings



This essential title in McGraw-Hill's *Robot DNA Series* is just what robotics hobbyists need to build an effective drive train using inexpensive, off-the-shelf parts. Leaving heavy-duty "tech speak" behind, the authors focus on the actual concepts and applications necessary to build — and understand — these critical, force-conveying systems. **$24.95**

CNC Robotics
by Geoff Williams



Now, for the first time, you can get complete directions for building a CNC workshop bot for a total cost of around $1,500.00. *CNC Robotics* gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot and can be customized to suit your purposes exactly, because you designed it. **$34.95**

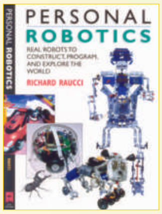
PIC Robotics: A Beginner's Guide to Robotics Projects Using the PIC Micro
by John Iovine

Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU! In this heavily-illustrated resource, the author provides plans and complete parts lists for 11 easy-to-build robots — each with a PICMicro brain. The expertly written coverage of the PIC Basic Computer makes programming a snap — and lots of fun. **$19.95**

Robots, Androids, and Animatrons, Second Edition
by John Iovine

There's never been a better time to explore the world of the nearly human. You get everything you need to create 12 exciting robotic projects using off-the-shelf products and workshop-built devices, including a complete parts list. Also ideal for anyone interested in electronic and motion control, this cult classic gives you the building blocks you need to go practically anywhere in robotics. **$19.95**

Personal Robotics: Real Robots to Construct, Program, and Explore the World
by Richard Raucci



Personal Robotics gives an overview of available robot products, ranging from simple to complex. Interested readers will be able to find the robot kit that matches their skill level and pocketbook. Other criteria a reader will be able to review include motion systems (from robot arms to robots that roll on wheels or walk on legs), available sensors (from none to a wide range), and programming complexity (how the robot is programmed). If it's really a robot, it's in this book. **$25.00**

Robot Builder's Bonanza
by Gordon McComb

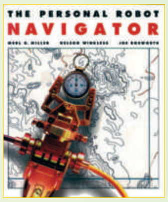
Robot Builder's Bonanza is a major revision of the bestselling bible of amateur robot building — packed with the latest in servo motor technology, microcontrolled robots, remote control, LEGO Mindstorms Kits, and other commercial kits. It gives electronics hobbyists fully illustrated plans for 11 complete robots, as well as all-new coverage of Robotix-based robots, LEGO Technic-based robots, Functionoids with LEGO Mindstorms, and location and motorized systems with servo motors. **$24.95**

Robot Builder's Sourcebook
by Gordon McComb

Fascinated by the world of robotics, but don't know how to tap into the incredible amount of information available on the subject? Clueless as to locating specific information on robotics? Want the names, addresses, phone numbers, and websites of companies that can supply the exact part, plan, kit, building material, programming language, operating system, computer system, or publication you've been searching for? Turn to the *Robot Builder's Sourcebook* — a unique clearinghouse of information that will open 2,500+ new doors and spark almost as many new ideas. **$24.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

The Personal Robot Navigator
by Merl K. Miller / Nelson B. Wrinkless / Joseph H. Bosworth / Kent Phelps

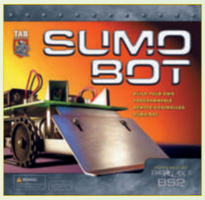


The Personal Robot Navigator is a book and software combination that presents the world of personal robot navigation — how robots can find their way around our homes and offices. The RoboNav® Simulation software included with the book enables you, the robotmaster, to establish a map for each room in your house, place furniture in the room — just as it is arranged in real life, record navigation corridors for the robot to travel, automatically calculate the best path from and to points in the room, simulate the robot following the path while monitoring the compass, sonar, wheel turn, and triangulation indicators on a sensor control panel, freeze the action and print the screen, and study the navigation log created by the robot. **$44.95**

PDA Robotics
by Doug Williams

The virtual chasm between PDAs and robots has been spanned with McGraw-Hill's *PDA Robotics: Using Your Personal Digital Assistant to Control Your Robot*, an easy-to-read guide to integrating these two pieces of technology into a single, remote controlled powerhouse. **$24.95**

SUMO BOT
by Myke Predko / Ben Wirz



Here's a fun and affordable way for hobbyists to take their robot building skills to the next level and be part of the hottest new craze in amateur robotics: Sumo competition.

Great for ages 14+, the kit comes complete with:

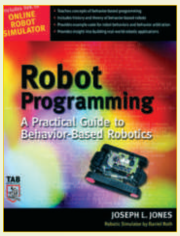
- Pre-assembled PCB
- Multi-function, dual-channel remote control
- Robot hardware, including collision-sensing infrared LED and receivers
- CD-ROM with programming instructions and file chapters of robot building tips and tricks
- A built-in Parallax BASIC Stamp 2 and prototyping area, allowing hobbyists to create their own robot applications without having to purchase additional Parallax products **$99.95**

To order call 1-800-783-4624 or go to our website at www.servomagazine.com

Build Your Own Humanoid Robots
by Karl Williams

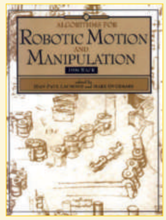
Build Your Own Humanoid Robots provides step-by-step directions for six exciting projects — each costing less than $300.00. Together, they form the essential ingredients for making your own humanoid robot. **$24.95**

Robot Programming
by Joe Jones / Daniel Roth



Using an intuitive method, *Robot Programming* deconstructs robot control into simple and distinct behaviors that are easy to program and debug for inexpensive microcontrollers with little memory. Once you've mastered programming your online bot, you can easily adapt your programs for use in physical robots. **$29.95**

Algorithms for Robotic Motion and Manipulation: WAFR 1996
Edited by Jean-Paul Laumond / Mark Overmars



This volume deals with core problems in robotics, like motion planning, sensor-based planning, manipulation, and assembly planning. It also discusses the application of robotics algorithms in other domains, such as molecular modeling, computer graphics, and image analysis. Topics include:

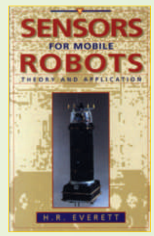
- Planning
- Sensor-Based Motion Planning
- Control and Motion Planning
- Geometric Algorithms
- Visibility
- Minimalism and Controllability
- Algorithms for Manufacturing
- Contact and Tolerance
- Beyond Robotics **$59.00**

Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.

The Ultimate Palm Robot
by Kevin Mukhar / Dave Johnson

Originally developed by Carnegie-Mellon University robotics department graduate students, this prototype has enjoyed a cult following among enthusiasts. Using software provided by the authors and this step-by-step guide, you can build and operate your own version of the same robot. Learn about parts, software, programming, games, robot resources, and much more from this exciting, one stop guide to Palm robots. **$29.99**

Sensors for Mobile Robots
by Hobart R. Everett



In *Sensors for Mobile Robots*, the author compiles everything a student or experienced developmental engineer needs to know about the supporting technologies associated with the rapidly evolving field of robotics. **$69.00**

CUTTING EDGE ROBOTICS

Part 5 — Combat Flipper

by John Myszkowski

We are making such great progress with our multi-function C. E. (Cutting Edge) Robot that I decided to temporarily change direction and throw caution to the wind. A fighting robot is what many of you want, so that's exactly what we are going to make: a Super Duper Scooper — the robot killing machine (Figure 1).

Building a robot is a learning experience that is hard to forget and the social aspect of competition is a welcome part of it, but fun is still the most important element of this hobby. I am constantly looking for the newest and best robotics experience or opportunity and, recently, I have investigated the offerings of the RFL (Robot Fighting League).

RFL is an open organization that promotes combat robot competitions with a variety of weight categories. Browsing through the different categories, I chanced upon the ANT weight, which is exactly where our C. E. Robot can fit.

I can see the obvious question forming in your mind, "Where's the weapon?" Well, this is the installment where we create an offensive weapon for our baby. If you decide to try out the combat side of robotics — even on a small scale — then visit the RFL website and check out the closest RFL club and fight times.

Before you even start thinking about combat, you first have to ask yourself if you can stand the possibility of your baby (robot) getting bashed into little pieces. I'd say go for it. You built it; you can rebuild it. You certainly have the technology and this will give you the excuse to make all the fixes and mods that you missed out on before. The templates allow you to make and carry many spare parts to replace any damaged ones.

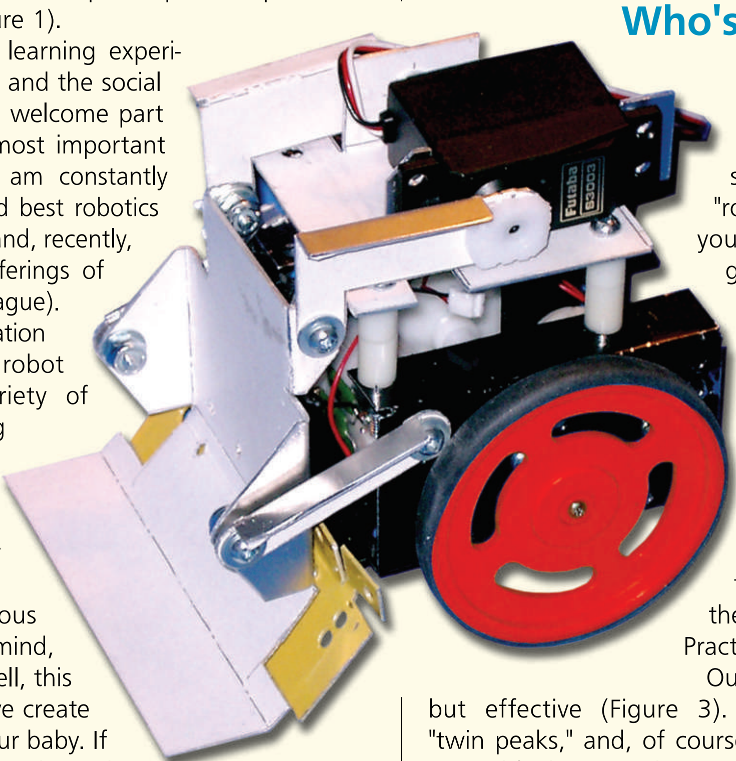


Figure 1. (above) Super Duper Scooper — the robot killing machine — or "lifting scoop," almost complete and ready to be decorated.

Who's on First

Through close observation and personal experience in robotics competition, one statement keeps popping up — "rock-paper-scissors" (Figure 2). If you are familiar with this old game, you will agree with the comparison. The simplicity of this game shows that weapons often play only a minor role and anyone can be the winner, depending on robot technology, circumstances, and experience. The PPP of competition is of extreme importance; if you want your robot to have even the slightest chance of winning, then you will need to Practice, Practice, and Practice.

Our robotic arsenal is quite simple, but effective (Figure 3). It consists of a flipper, the "twin peaks," and, of course, improved wheel traction. The unmodified C. E. Robot can be flipped and left defenseless on its side, but the "twin peaks" will reduce that possibility. The increased wheel traction will help to focus power into pushing against the floor, rather than spinning in place.

Oh, How Tacky

Traction and the casting of custom polyurethane rubber tires have already been covered in a previous issue of *SERVO*. I would like to emphasize that wheel traction can sometimes

be the only deciding factor in a robotic combat. I will not even try to summarize these articles, but I do suggest that you get a hold of all the previous issues of *SERVO* while back issues are still available.

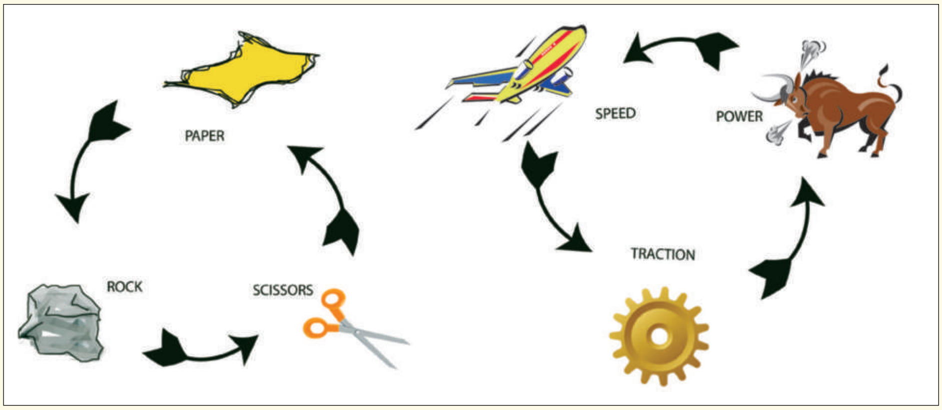


Figure 2. The winning cycle — "Rock-paper-scissors" — is the prevailing factor in winning and, in robotic competition, it translates to "speed-traction-power."

Flipping Out

Okay, okay. Our weapon may not look extremely destructive, but it is offensive, as well as defensive. We could add a saw or a spinner at some later date, but the flipper is much more than just a weapon. It is the base plate for our ARM (Advanced Robotic Manipulator) that we'll eventually get to build (Figure 4). It is also the lifting platform for our gripper (hand) and the camera (vision) that we will add in upcoming installments.

The flipper is made up of six parts: the base, four lifting arms, and the lifting scoop (Figure 5). The mechanical power comes from an R/C (remote control) servo mounted on top of the base. Each lifting arm is made of two pieces to form a parallelogram that keeps the lifting platform parallel to the ground. The material can be plastic, but I strongly suggest 1 mm thick aluminum for its rigidity and ease of handling.

Token Safety Message

I mentioned this before and I know many of you skip the safety talk, but it is essential that you put safety first. Safe working habits will help you keep enjoying the rest of your life. So ... *safety, safety, safety*!

Remember that cutting tools are not prejudiced — they will try to cut metal, wood, or your hand with equal vigor. You are the only one who can prevent accidents and possible injury. Always use PPE (Personal Protective Equipment) when working with cutting tools.

That said, let's get back to work.

Constructive Thoughts

Just as in our previous projects, we need to gather all of the necessary materials and tools. All of the metal parts will fit on a single 8.5" x 11" sheet of aluminum and no new tools are necessary (Figure 6). Simple, hand-held sheet metal snips should be more than sufficient, but a band saw is strongly suggested. Any bending can be done with a vise and/or a couple of pieces of scrap wood, but I suggest a small brake that can be purchased for about $20.00 from just about any tool supplier.

The template has to be printed 1:1 (100%) using the file on the website; it can also be photocopied from the magazine. After printing, make sure the reference dimensions are correct by measuring them. The closer your actual dimensions are, the less fiddling will be needed later.

Sticking the paper template to the sheet metal is a simple process. I use a cheap glue stick, but you may use a

Figure 3. Three main components of our combat robot include: flipper/scoop, cone wheel protectors, and better traction through molded soft rubber tires.

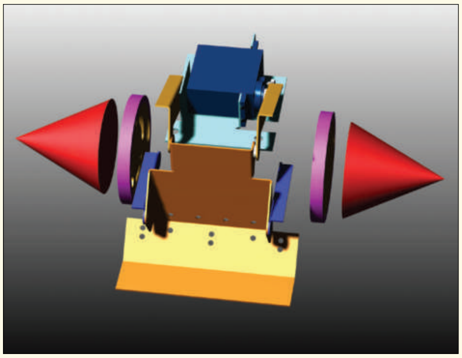


Figure 4. The ARM — this photo is worth a thousand words. The ARM, which we will build in a future article, is an extension of the Cutting Edge Robot.

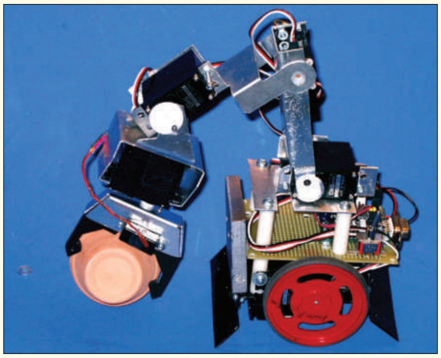
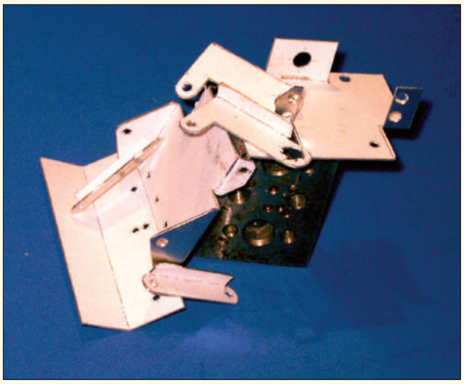


Figure 5. All the flipper/scoop components have been cut and bent and are awaiting assembly.



A Multi Function Robot — Part 5

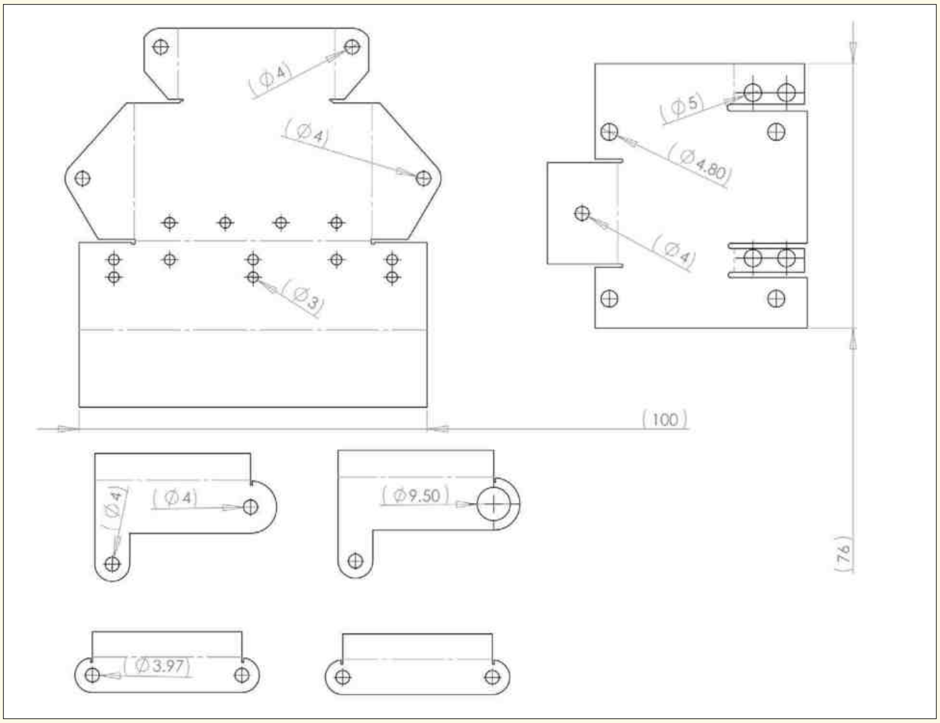


Figure 6. These are the necessary pieces for our project. Just add metal, muscle, and some tools.

more expensive one. Make sure the templates will not slip or come off during fabrication. The center marks and the broken "bend" lines are critical for proper component alignment and operation. The procedure is the same as before, but I will quickly reiterate the sequence:

- Use a center punch to mark the hole centers.

- Drill all of the holes first.

- Cut out the parts from the metal sheet.

- Bend where necessary, as indicated on the template.

- Deburr the sharp edges with a de-burring tool and/or a small file. A small countersink bit is great for deburring the holes.

- If you plan to paint the mechanism, first assemble everything together and make sure it works. After you are happy with its operation, take apart the flipper and prepare for painting.

Operation

The most important fabricated part of the flipper is the base plate that mounts on the C. E. Robot's chassis and holds the R/C servo (Figure 7). The servo is the power component that does the actual lifting. The base, arms, and flipper scoop form a parallelogram. The top pair of arms is moved by the servo and it moves the scoop up or down. The bottom pair keeps the scoop body parallel to its original position.

Now Serving

Depending on the servo, you should be able to easily lift a one pound load with the business end of the flipper scoop. Servos that are rated at 72 in-oz torque roughly translate to an ability to lift four pound with a one inch long arm, but, if you lengthen the arm to four inches, you will need to divide the torque by four. In other words, the Total-Torque equals the Rated-Torque divided by the Length (Figure 8).

TT=RT/L

Figure 7. The main component of the flipper/scoop is the base — used later for mounting the gripper, camera, and even the ARM.

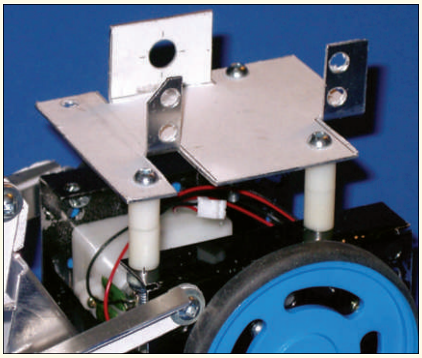


Figure 8. The torque of a servo mechanism or a gear-motor is limited by the specified rating.

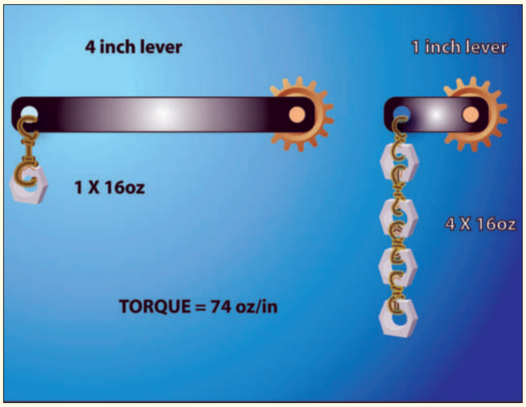
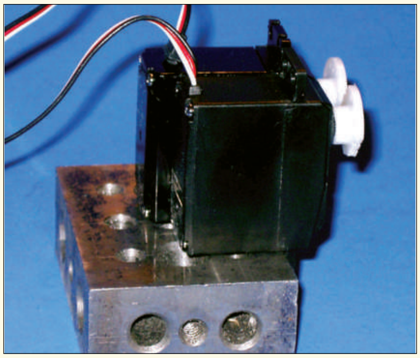


Figure 9. R/C servos are sized similarly for mounting purposes, except for depth. Compared here are a smaller 60 in-oz servo and a larger 100 in-oz unit.



It's not a precise formula, but it will work in our case. The length (L) is the length of one of the top arms plus the distance from the vertical wall of the flipper to the front end of the flipper scoop. It should be approximately four inches. I suggest getting a stronger servo — something in the range of 100 in-oz of rated torque.

The only limiting factor is the size, so a GWS-06 (or equivalent) will probably be the largest servo that will fit with some fiddling. My choice was the GWS-03BB. It is capable of about 70 in-oz and has ball bearings at the shaft that will probably extend the working life (Figure 9).

Poop on the Scoop

The most effective mode of flipper operation is flipping the opposition, but you will first have to get under them. Two factors seem very important:

1. The scoop's front edge needs to be touching the arena floor.

2. The same edge (we can call it the Cutting Edge) needs to be sharp enough to cut.

By making the hinges a little loose or sloppy, we allow the flipper scoop to follow the arena floor, which meets the first requirement. The Cutting Edge of the scoop is made with a file. Yes, we file the front edge until it slips under even the tightest armor (Figure 10).

Madonna Was Here

Being tall, the C. E. Robot is vulnerable to being flipped on its side, so we need to add some protection (Figure 11). "Twin Peaks" is the name I thought would be appropriate for the wheel protectors. These shields perform three functions:

1. Protect the wheels from spinning weapons.

2. Prevent the robot from falling sideways and spinning helplessly.

3. They look cool.

The pointy, cone shape is very easy to make and you can just hot glue it to your wheels. As always, painting or decorating is optional, but, the better your fighting machine looks, the better it will be liked by the audience — sometimes, that's all that counts.

Power Play

Our small fighting machine may not be the strongest or fastest kid on the block and, even with plenty of traction and sufficient mass, your robot may still need some extra speed to compete with the bigger boys. The GM2 motors have a close relative that is a little faster — the GM8 (Figure 12). This motor will provide more speed with a simple substitution, if that's what you want. The only problem with the GM8 motor is its short shaft, which will bring the wheels too close to the chassis. That doesn't leave enough room for mounting the bottom set of lifter arms.

In many cases, what your drive system needs is simply more electrical power — more current. The L293 motor drivers can supply plenty of power to the small DC motors, but they do have a limit. If they get too hot, they simply shut down until they cool off. In a furious fight — where two robots are fighting head to head and the motors are blocked and can't move forward — the current skyrockets and that may cause the L293 to momentarily shut down.

The power handling of the L293 chip can be almost doubled if you simply solder a second, identical chip in parallel with the first one. You can position the second chip on top of the first chip in a piggy-back fashion and solder the corresponding leads together (Figure 13). You should also place a heatsink on top of the IC, which will draw the heat away faster.

If you plan to use a larger chassis to build a bigger robot with even stronger motors, you may want to consider a stronger motor driver, such as the L298 H-bridge chip. The L298

Figure 10. The business edge of the flipper/scoop is sharpened with a file to be able to slip under the enemy robot before lifting it.

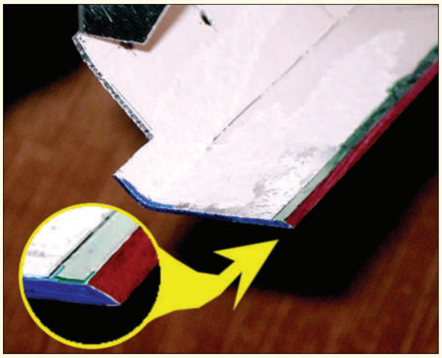


Figure 11. These cones of terror keep the robot from staying on its side if it has been flipped over.

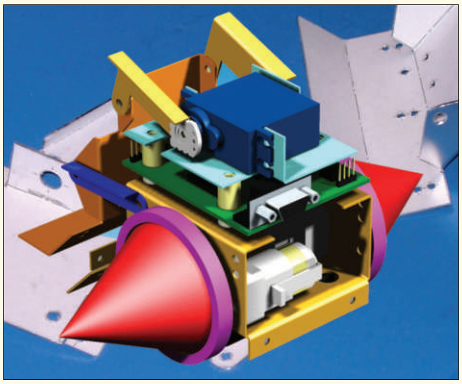
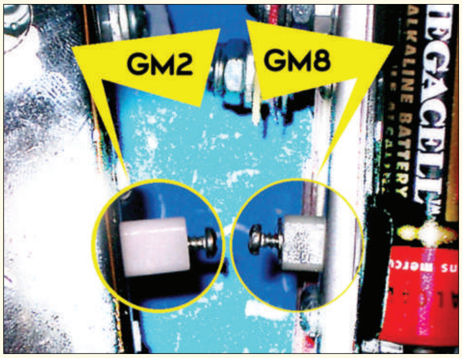


Figure 12. The size of the output shaft differs between different gear-motors. This GM8 motor has advantages over the GM2, but the shorter shaft makes it more difficult to use.



References

Books:
I have these books and I think they are the best for the budding roboticist:

- *Junkbots, Bugbots, & Bots on Wheels*
  ISBN 0-07-222601-3
- *Building Robot Drive Trains*
  ISBN 0-07-140850-9

Check out the following Internet links. They are all related to the Cutting Edge projects.

Supplier and Information Web Links:

www.novarobotics.com
C. E. Robot parts and custom CNC work.

www.hobbyengineering.com
GM2 motors and many different robot components.

www.1sorc.com
Sumo-11 boards and information.

www.basicx.com
Home of the BX-24.

www.cuttingedgeprojects.com
This is the place to check for more projects.

www.robotgames.ca
Robot information and competition.

www.robotgames.com
Robot information and competition.

www.solarbotics.com
Source of GM2 motors and other robot parts.

www.barello.net
R/C robot controllers.

www.bugnbots.com
Robot parts, GM2 motors, PCBs, etc.

www.junun.org
Mark-III robot, R/C servos, and other useful robot parts.

www.avrfreaks.net
AVR and everything you ever wanted to know about it.

www.ridgesoft.com
Java support for the Sumo-11.

www.botleague.com
Home of the Robot Fighting League and your gateway to the world of robotic combat.

www.servocity.com
R/C equipment, R/C servos, servo components, and useful information.

Yahoo User Groups:

http://groups.yahoo.com/group/CuttingEdgeProjects

http://groups.yahoo.com/group/MiniSumoMarkIII/

http://groups.yahoo.com/group/Sumo11users/

http://groups.yahoo.com/group/tabrobotkit

can pump out up to 3 A with sufficient heat dissipation and it can also be paralleled for even more power. The simple circuit configuration also includes the diodes necessary for noise reduction and negative spike elimination.

A different solution is needed for the cheap gear motors that were pulled out from battery-powered hand tools that draw huge amounts of current. The battery-powered screwdriver can draw as much as 10 A — even more under heavy loads or stall conditions.

What is needed is a discrete H-bridge circuit made from high power transistors that can take even the highest current spikes. The nice thing about discrete components is their low price and high availability. When they "pop," they are cheap and easy to replace (Figure 14).

Flea Power

At the other end of the spectrum, we have the micro and nano size robots (Figure 15). Their power requirements are so low that some of

Figure 13. Left to right: A single L293 chip can supply 1 A per motor, "piggy-back" dual L293s can supply almost 2 A, and the larger L298 is supposed to give you almost 3 A when properly cooled.

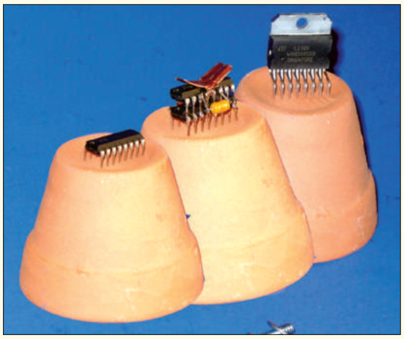


Figure 14. Are you ready for a full-sized Japanese style sumo? This C. E. Sumo is made from a single sheet of aluminum and uses two screwdriver motors.

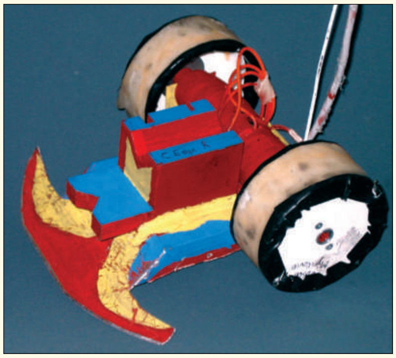
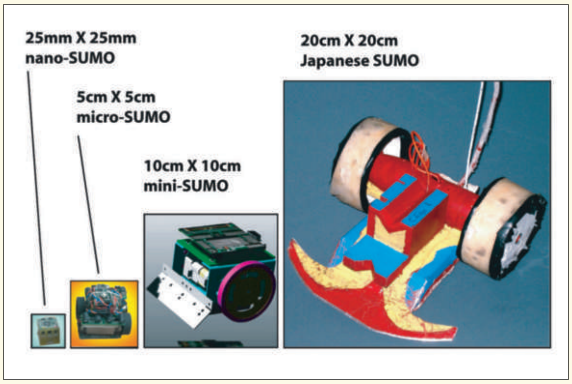


Figure 15. We can grow or we can shrink, either way is a welcome challenge. Left to right: Cyclops, Bob, Cutting Edge, and C. E. Sumo.



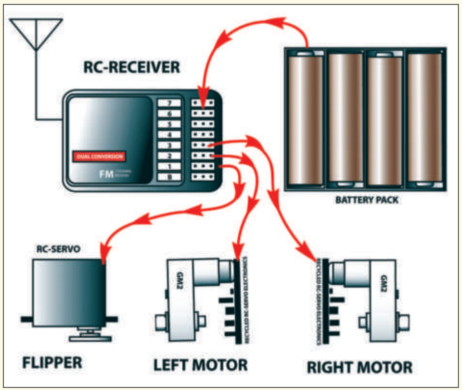


Figure 16. Hookup diagram of an R/C robot control system. Here is a simple setup with modified GM2 motors and recycled servo electronics.

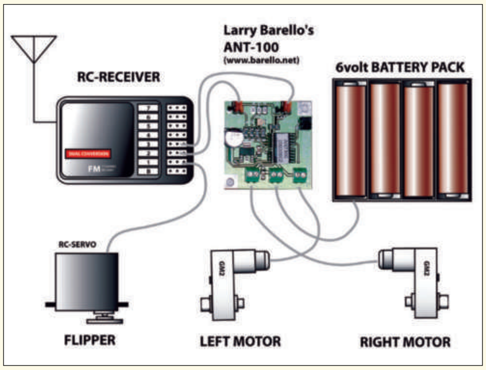


Figure 17. Other than recycled servo electronics, you can use a dedicated combat robot controller, such as the ANT-100, which can be used with a regular R/C rig and DC motors.

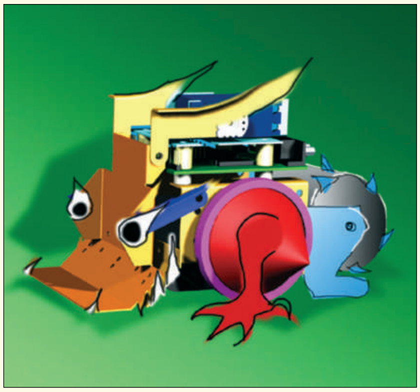


Figure 18. Coming next month — the assembly details of the flipper/scoop and addition of "Tail Spinner" to our combat robot.

the tiny motors can be driven (almost) directly by the microcontroller chip. It is extremely important to be able to understand manufacturer's specifications in order to successfully interface inductive loads to the micro. Personally, I would rather err of the side of caution and use a buffer between the micro and the motor — even if it does mean an extra chip or a transistor.

Control

In combat robotics, we have a choice of using R/C, full autonomous, or a combination. The RFL style of fighting in the ANT division calls for remote controlled operation and the rules specify the acceptable types of remotes. You have to abide by these rules for many reasons, but safety is the major one. The C. E. Robot can work within these specifications and even leaves us with a variety of choices concerning the controller.

For remote operation, we can use the radio receiver and reclaimed R/C servo electronics to control our robot (Figure 16). You can read the full R/C servo "reclamation" procedure in previous installments of "Cutting Edge" (back issues of *SERVO* are still available). The flipper/scoop is powered by an unmodified R/C servo which plugs into one of the receiver channels. You may want to add a flipper activator control in the R/C transmitter — such as a push-button switch for quick response. You can also just use one of the joysticks to activate the servo.

If you do not have the spare or broken servos ready to be recycled, the other option is to use a servo translation and motor driver circuit. Larry Barello (**www.barello.net**) has a number of circuit boards available for exactly that purpose at a very reasonable price (Figure 17).

See the reference listing for more choices for translation motor drivers.

Tail Spin

Before we get back on the track to our final destination — the ARM — we need to finish the assembly of the flipper/scoop and get a little deeper into the remote control aspects of our combat robot. We will build "Twin Peaks" wheel protectors and, if we have the room, we may even throw in a second weapon — "Tail Spinner" (Figure 18). **SV**

TETSUJIN TECH

Part 2 INTRODUCTION TO PNEUMATICS

by Alexander Rose

Pneumatics (the use of pressurized gas — usually air — to do work) has been used in automation and power transmission since Otto von Guericke first improved air compressors around 1650. Pneumatic tools started to gain popularity after Ingersoll Rand came out with their air powered drill in 1871 and Brady King developed his pneumatic hammer in 1890.

Since then, pneumatics has been radically developed and found its way into just about every industry, with varying degrees of success. Robotics showcases some of the most interesting modern uses of pneumatics, as do areas where electronics have been banned, such as Amish communities and NHRA stock car racing. In this — the first of two articles — I will detail some of the reasons you may want to use pneumatics in robots.

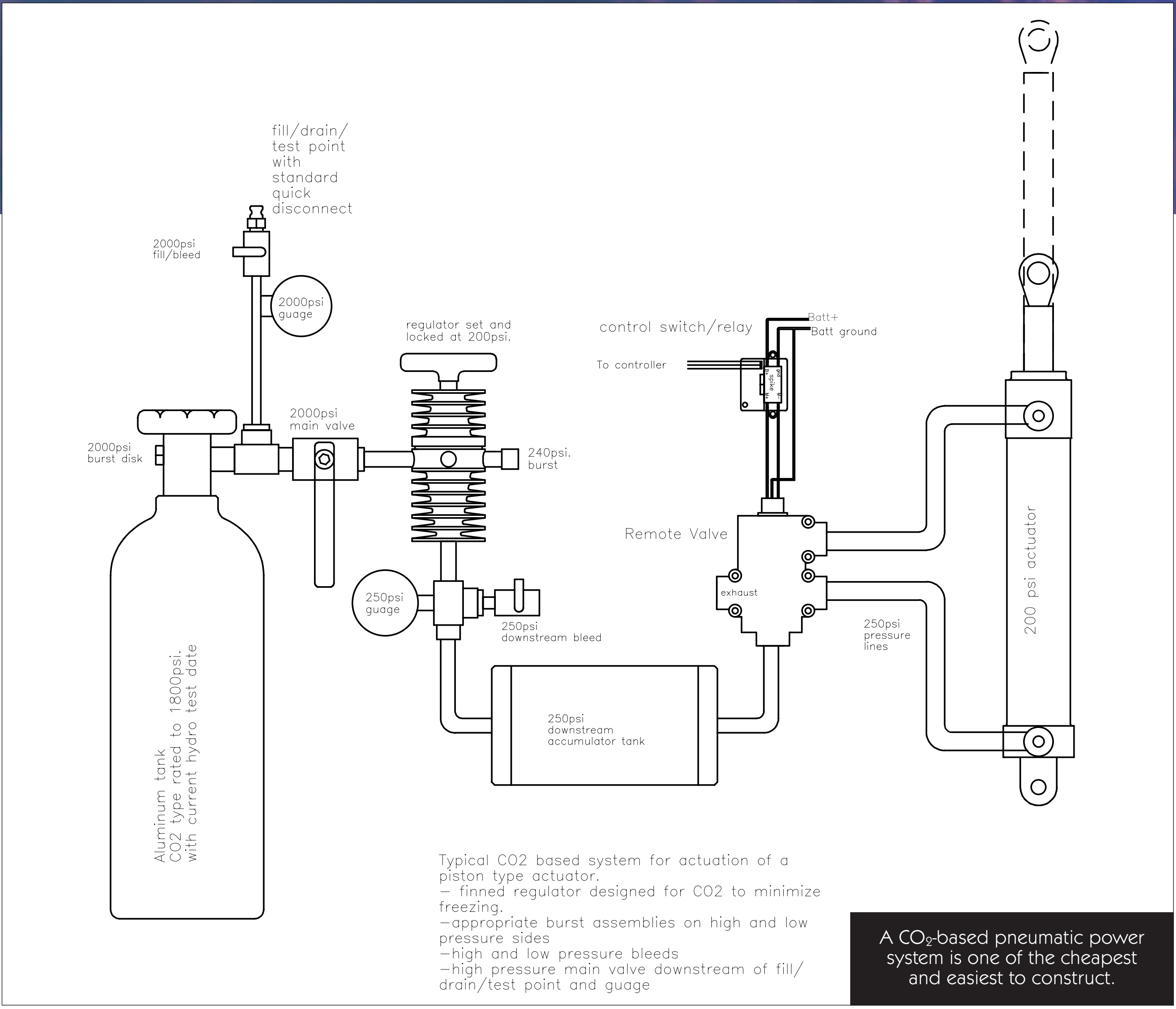
Using compressed gas to perform work has advantages and drawbacks — just as electricity does — and these should be evaluated before heading down the pneumatics road. In general, the best tasks for pneumatics are ones where you need short, defined bursts of high power and speed. Some examples include lifting, flipping, clamping, and fast rotation.

Pneumatics is usually a bad choice for continuous rotation or any type of motion that requires fine modulation. Most pneumatic systems are binary in their function; they are either on or off. As with any rule, there are exceptions and hacks to this, which I will cover shortly. One of the other key benefits to using pressurized gas is the ability to vary its force or speed very easily through pressure and orifice restriction.

Failure in pneumatic systems is similar to that of electronic systems — fast and catastrophic. Pneumatics has the added danger that, when it fails, parts may explode and become projectiles or very powerful whips. When working with pneumatics, it is always wise to take proper safety precautions and never exceed the pressure ratings on components. The danger in a pneumatic system goes up exponentially as the pressure increases. Take advantage of the fact that you can test your system at very low pressures first, then gradually increase to working pressure as you test.

The author's nitrogen charged "fire fountain" of kerosene at the Burning Man art festival.

One of the critical factors worth evaluating before using pneumatics is whether or not you want to have two whole systems onboard your robot. In general, there is no such thing as a pneumatics-only robot; in robotics, pneumatics is almost always paired with electronics. This simple fact means that, by using pneumatics in your robot, you not only have to worry about the electronic power source and transmission



A CO_2-based pneumatic power system is one of the cheapest and easiest to construct.

The author lines up for a run at the “power tool drag races” in Northern California.

system, but also the source of gas and all of the componentry.

The task should be weighed carefully to see if using this second system in your robot is really worth it. We used pneumatics in a fighting robot that needed to exert over 10,000 pounds of lifting force in .01 of a second. This application was ideally suited to pneumatics and would have been very difficult to emulate with electronics.

When designing a pneumatic system, you should look at the task you want to accomplish and decide how much force and speed are needed. In many competition applications, the requirements are usually, “as fast and strong as possible.” In this case, most of your research will go into componentry.

The two main points of restriction in any pneumatic system are generally the regulator and the valve. There are ways around these restrictions — such as the use of buffer tanks and the use of valves in parallel — but, in general, you will want to find the regulator and valve with the most throughput.

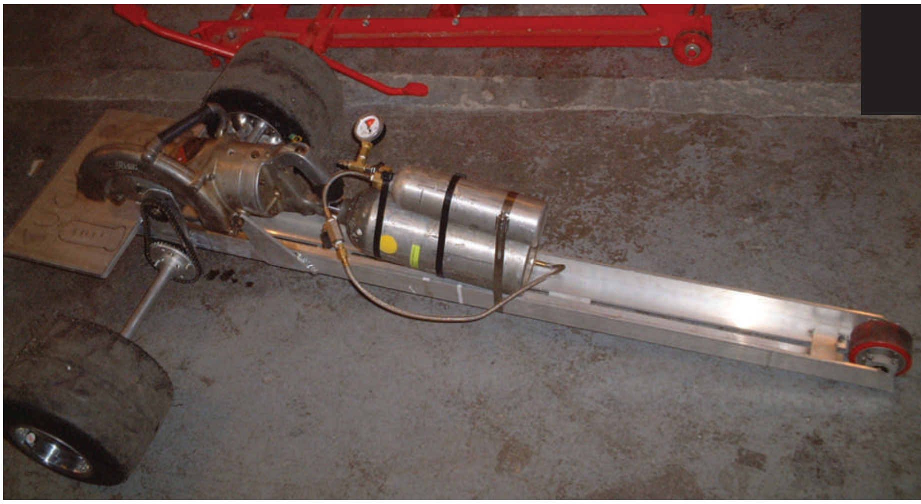
How much is enough? You can calculate the theoretical force of your system by multiplying the area of your actuator times the pressure, but this is usually just the beginning in terms of understanding how your system will perform. Furthermore, calculating gas volume and speed requirements is a vague science. Since the number of variables is high (line friction, right angles, temperature, etc.), it is usually easiest to make a simple test setup first.

You will have to invest in some basic components; most vendors will have personnel to help you select the best parts for your application. You will quickly learn the capabilities and shortcomings of your system this way and determine which components you would like to improve.

When you come to the point of choosing your final components, you will also begin to have an idea of how much operating pressure and speed your application will require. In the US, units are referenced as follows: pounds per square inch (psi) is pressure, cubic feet per minute (cfm) is gas consumption, and cubic feet (cu) is gas volume.

The pressure of your system is more often determined by commercially available components than anything else. Most of the off-the-shelf actuators, tanks, and valves operate between 20-300 psi, with the vast majority of them being rated no higher than 150 psi.

Choosing the cfm of your system is also largely at the mercy of componentry; in general, you want the largest cfm possible for fast moving systems; you can always artificially restrict it later. The speed of your system will be based on its slowest (most restrictive) component. The pressure your system is operating at



In the shop — the compressed air powered dragster is built from a modified pneumatic circular saw.

— its cfm — and component size are the elements required to derive the total volume of gas required.

Once you know the pressure, speed, and volume requirements, you will want to decide on your source of compressed gas and what gas to use. In general, there are three types of gas sources used in robotics and industry: low pressure compressed air from a simple compressor, high pressure gas from a tank, or liquefied compressed gas from a tank.

For applications where the robot can be tethered or will require a constantly regenerating source of air, the use of a compressor is often easiest. For applications where the robot only works for a limited time and can be serviced in between tasks, an onboard tank is generally used.

In the next issue, we will cover gas source types, componentry, and system optimization. Until then, start planning pneumatics for your Tetsujin exoskeleton and stay tuned. **SV**

About the Author

Alexander's formal background is in industrial design. He began working with pneumatics in the paintball industry in 1987. Since then, he has used pneumatics successfully in championship fighting robots, a dragster made from an air powered circular saw, 300-foot-tall flamethrowers, and, most recently, one pound micro fighting robots.

Bot Builder + Couch Potato

United Through the Archos Remote

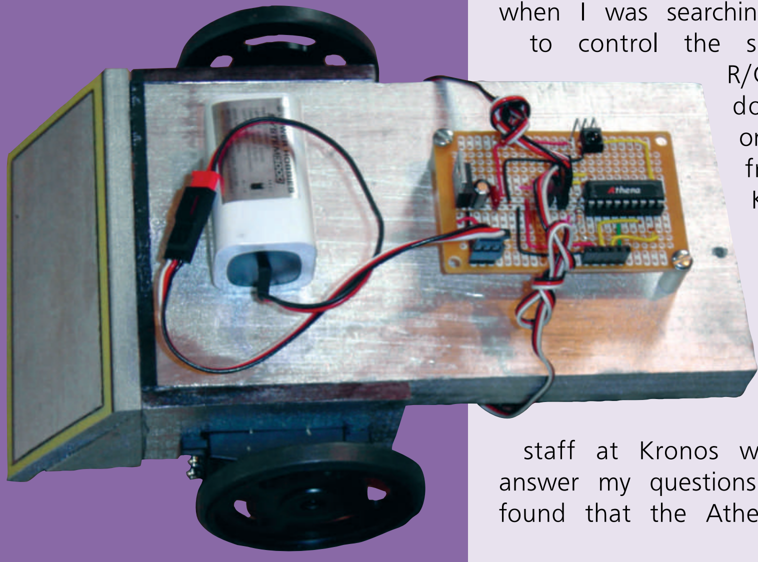
by Bob Knoblauch

With the Athena microcontroller, I built my own IR (infrared) controlled robot. The bot may not move very quickly, but it is extremely quiet; it's stealthy enough to sneak up on the family cat!

The Athena microcontroller is easy to learn and can be purchased for a very reasonable price.

Playing with the cat is a fun bonus, but building the bot was a learning experience for me in many ways. I figured out how to reprogram our universal TV remote control to send IR commands through an IR module to the Athena microcontroller. Then, using the Athena chip, I was able to control the bot's R/C (remote control) servos. (My son had some spare servos from one of his remote control planes that I borrowed to build my robot.)

One Good Hobby Leads to Another ...



I discovered the Athena chip when I was searching for a way to control the servos in an R/C plane. The documentation on the chip was free on the Kronos website (**www.kronosrobotics.com**) and I found it easy to comprehend. The staff at Kronos was happy to answer my questions via Email. I found that the Athena chip was already set up to talk to the R/C servo and had open I/O ports that I could use for other things. It's programmed in Basic and the language is a free download from Kronos' website. As noted, the microcontroller is inexpensive, too.

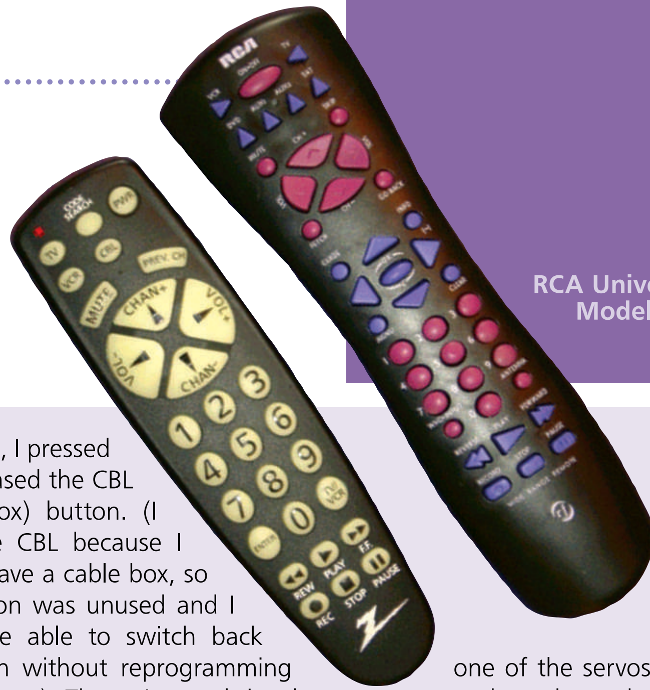
Before using the Athena chip, you will need to build a port adapter. The port adapter converts the voltage to the correct levels for the Athena chip.

I was able to order an inexpensive kit from Kronos Robotics — the EZ232. The kit is indeed "EZ" to build with and it will be useful for other building projects, which is an added bonus.

The next step is to set up the Athena on a breadboard and wire it in. I followed the example that came with the manual.I became more familiar with the Athena by using it to play with LEDs — programming them to turn on and off in different patterns. I just added LEDs to the ports and hooked all of them up to a 390 Ω resistor that went to the +5 volt power source.

Then, I borrowed the R/C servo from my son and used the Athena to make the servo move. I discovered that the Athena could be programmed to understand the IR controller that is made to control a Sony TV. (My controller wasn't a Sony, but I reprogrammed it to act like a Sony.) My son was glad to lend me the R/C servo, but he protested the absence of the TV remote. I promised him that he could have it back when I was done playing.

Continuing to look around on the Kronos website, I discovered that you can modify the servos to turn 360°. Instead of modifying the servo, I just



Left:
Zenith ZEN755

Right:
RCA Universal Remote Model D770 Rev. B

ordered two servos that were already modified.

I also noted that Kronos featured a bot similar to the one that I wanted to build. The one featured on their site is shown in two versions. One version has two servos — like the bot I was making — and the other version has been modified with sensors to be able to detect objects. I decided not to use the object-detecting sensors for my bot, since I would be controlling it with the TV remote. I did use some of the same methods for my bot program that Kronos had used for their bot, though.

Programming the Zenith Remote (ZEN755)

I looked up the TV code for Sony in the Zenith remote manual that comes with the remote. The manual has two different Sony codes listed. I tried the first code and it didn't work, so I tried the second code, typing 801 with the number keys. This code worked.

I could tell that the 801 code was working, even though I had tested this particular remote before I built my bot. I had set up the Athena chip to echo the information back to the computer.

On my first try, nothing happened, but, the second time, the Athena chip echoed back the numbers.

First, I pressed and held the code search button until the indicator LED lit up, then I released the code search button.

Next, I pressed and released the CBL (cable box) button. (I used the CBL because I do not have a cable box, so the button was unused and I would be able to switch back and forth without reprogramming the remote.) Then, I typed in the code.

Programming the RCA Universal Remote (Model D770 Rev. B)

To program the AUX1 and AUX2 manually on an RCA remote, the process is a bit different.

First, press and hold either of the two AUX keys. Then, using the number keys on the remote, press 1001. This is the Sony TV code. (A list of other codes come with most programmable remotes or you can visit **www.rca.com** to find out more information.)

If the first number of the code is valid, then the LED will turn off. After you've entered the full code, the LED will turn on to verify the code.

Bot Building

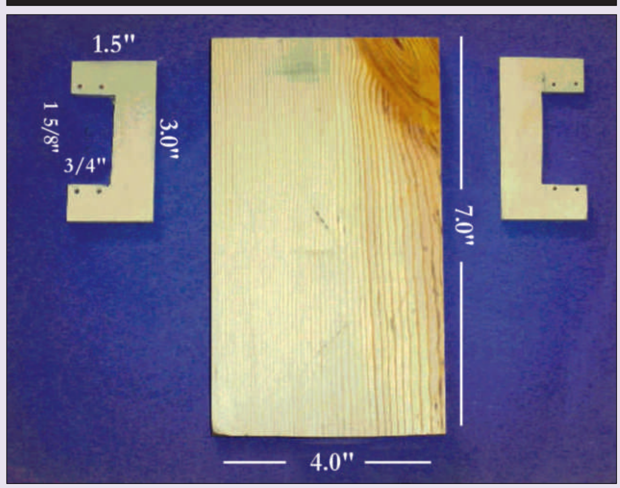
Once I felt I was reasonably proficient in controlling the servos, I started building my bot. I used a 1/4 inch thick piece of wood and cut it to 3 x 1.5 inches.

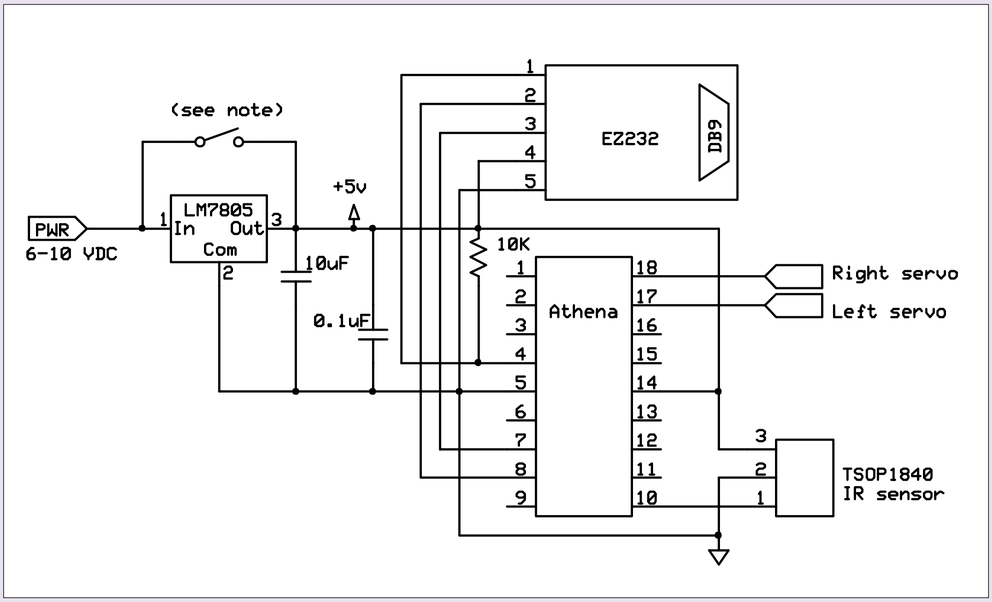
Then I put one of the servos next to the wood and marked out the opening for it as a cutting guide. I did this again for the other servo.

Next, I drilled guide holes where the servos would go. I used the holes on the servo to line up the drill and then attached everything together. The two servos were mounted on opposite sides of the bot's body with their bases facing each other.

For the bot's body, I used a 3/4 inch piece of wood. I cut this to 7 x 4 inches. I glued together the two pieces that would hold the servos onto the main piece of wood. They were glued to the edge of the 7 inch side.

The wooden base and servo mounting ears.





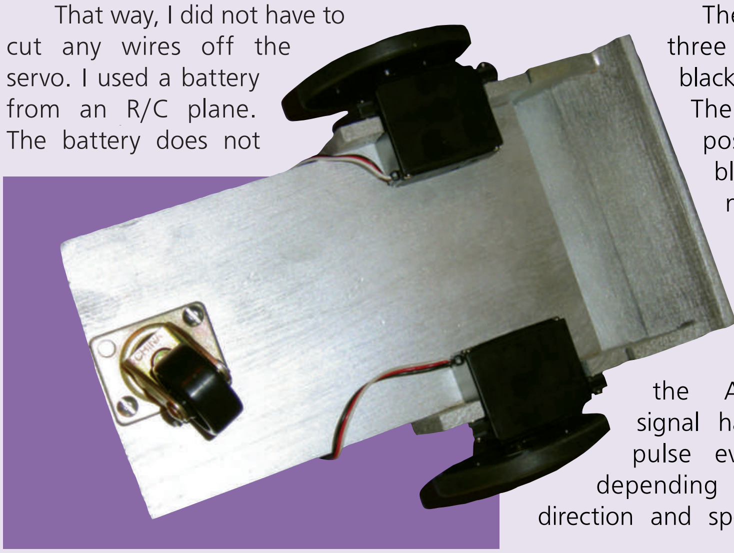
A simple schematic: voltage regulator, Athena controller, IR sensor, and RS232 programming adaptor.

Finally, I mounted the caster wheel in the back. My first set of wheels for the servos was a pair of plastic peanut butter jar lids. Later, I replaced the lids with foam wheels designed for the servos.

After the bot was built, I transferred the electronic components from the breadboard to a PC board. I used an 18-pin socket for the Athena. For the components that weren't on the PC board, I used headers so that everything plugged into the PC board. It works out well to plug the servo into the male header.

That way, I did not have to cut any wires off the servo. I used a battery from an R/C plane. The battery does not have to be secured on the bot, since it does not go fast enough to make the battery fall off. The battery plugged into the male header, too.

It's not necessary to use an R/C battery. Four AA batteries can be placed in a battery clip and attached. I plugged the EZ232 into the female header.



No Guts, No Glory …

Here is more detailed information on the servos, the IR module, the voltage regulator, and the Athena itself:

The servo has three wires: red, black, and white. The red is the positive, the black is the negative (or ground), and the white is for the signal that comes from the Athena. The signal has a 1-2 ms pulse every 20 ms, depending on the direction and speed that you want the servo to go. The Athena chip has built-in commands to do all the work of talking to the servo.

The Vishay IR module is a three-pin device. Pin 1 is the output, pin 2 is ground, and pin 3 is for the +5 volt output. Looking at the top (with the dome up) the pins are (from left to right) 1, 2, and 3.

The 7805 is a 5 volt, 1 amp regulator. Looking at the face of the regulator (the print is on the face), the pins are 1 for input, 3 for ground, and 2 for the 5 volt output. Depending where you buy the regulator, the pins may be numbered in a different order, but the input, output, and ground will always be in the same position.

The Athena has a dot for pin 1 and is an 18-pin chip. I used a DIP socket to avoid damage to the chip while I soldered the PC board. For this bot, I used three I/O ports — one for the IR module — pin 10 (port number 5) and two for the servos. The left servo is on pin 17 (port number 1) and the right servo is on pin 18 (port number 0.)

There are 12 I/O ports left. Two of the ports that are used for programming the chip can be used for other types of communication, once the Athena chip has been programmed.

"Hi-ho Silver!"

After I had put the bot together, I tried it out to make sure that everything was working right. Then I took it apart to paint it. I used silver paint for the body and accented it with yellow and black lines.

I also decided to give the bot a plow-shaped bumper, but that is optional.

How It Works

You either point the TV remote

at the bot or bounce the signal off something reflective, such as a mirror. The IR module receives the information from the remote and the Athena microcontroller decodes it.

Depending on the information, it will turn the servos in different patterns. For forward (the channel up button), the Athena chip turns both servos.

Because the servos are mounted back-to-back, one will be turning clockwise and the other counter clockwise. To go backward, use the channel down button. The volume up/down buttons will control left and right turns.

The most common and widely used TV remote control is the IR remote. It works by using an IR light beam that cannot be detected by the human eye, but is detected by a receiver in the TV (or bot, in this case.) The information is sent out of the remote in a pulsed light beam. The effect is similar to the RS232, but, instead of voltages, it is in a light format.

To turn right, both servos will turn clockwise. To turn left, both will turn counterclockwise. If you hit the forward button and the bot goes backward, you can just swap the plugs that the servos are plugged into.

If the bot turns left when you wanted it to turn right, you can just change the code.

This way, you are not restricted to putting in the servos in precisely the same way that I did. **SV**

Parts List

Electronics

Athena	Athena microcontroller (16276)
EZ232	Easy RS232 interface kit (16167)
IR1	TSOP1840 IR sensor (16226)
R1	10K, 1/4W resistor
C1	0.1µF 50V capacitor
C2	10µF 50V capacitor
VR1	7805 5V regulator
B1	4.8V or 6V NiCad pack (see notes)

Mechanics

Drive motor	Modified servo (16318)
Wheels	Servo wheels (16319)

Misc.

36 pin male and female headers, 18-pin DIP socket, six foot 9-pin serial male to female cable (16259), 1-1/4" caster wheel (pair), wood, paint, etc.

Notes

- When using a 4.8V NiCad pack, you must jumper over the 5V regulator (the SPST switch in the schematic).
- Parts available from Kronos Robotics (**www.kronosrobotics.com**) are followed by their product ID in parenthesis.

Composite Techniques *for* Robot Builders

by Tom Brusehaver

Composite construction sounds really high tech and complicated. Actually, the word composite just means that a material is composed of various other materials. Military aircraft and snowboards made from carbon fiber are two applications of composites that come to mind. They are constructed using techninques similar to what I will describe here.

Robots — just like airplanes and cars — benefit from being strong, yet lightweight. Batteries will last longer and the robot won't need so many. This means smaller motors, lighter wire, and a more efficient robot.

When most people first think about fiberglass or composite products, they usually think about the finished product — a bit of printed circuit board for instance, or a boat. Fiberglass parts are composed of three components: the cloth, a binder, and a core material. The reality is that composite structures incorporate several materials.

The cloth can be fiberglass, Kevlar, carbon fiber, or another material. The binder can be epoxy, polyester, or vinylester resins. The core can be more glass, wood, paper, foam, or aluminum.

Cloth

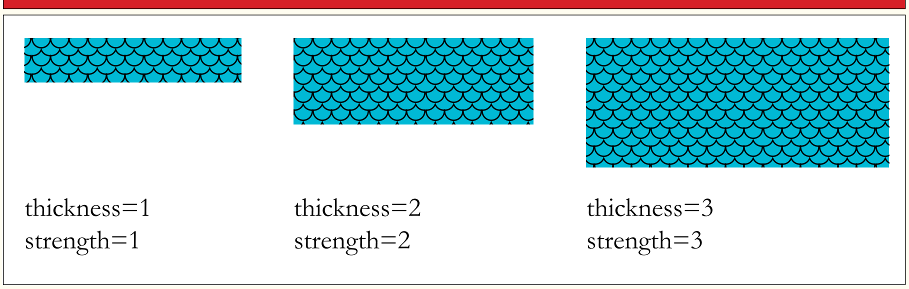
The cloth is available in a wide variety of types, weaves, and materials. Standard fiberglass — like the kind PC boards use — is an "E" or electrical glass. A little stronger version is called "S" glass. "T" and "A" types are also available. Carbon fiber is simply a cloth made with a high content of carbon and it is stronger than the "E" or "S" types of glass. There are also some exotic cloths, including Kevlar, ceramic, nextel, and spectra.

Most cloth will come with a sizing or finish. This finish (such as Volan-A) is a coating designed to help the resin penetrate the cloth. The finishes wash off easily. If the glass gets wet, the finish will be damaged, making it a poor candidate for any strength layups. If a bargain cloth is obtained, it will likely lack a finish or have an improper one, so it should also be avoided in cases where high strength is desired.

Fiberglass is really just strands of glass. The cloth is made from roving, which is what strands of material are called in this application. These rovings can be woven, used straight, or chopped. Many boats and high volume fiberglass parts are made from chopped roving that is applied using a gun; this is fine for factory work, but isn't as strong as woven glass cloth per pound of finished product. A hand layup can be the lightest and strongest construction method available to the robot builder.

Cloth is available in various weaves. The weave can be a straight weave (one up, one down) or a satin weave (two to five up and down), which is usually called bidirectional (or bid) and is equally strong in both directions. Unidirectional (or uni) weave — heavy roving mostly in one direction with some binding roving across it — gives the most strength in the direction of the long roving. There are knitted weaves which use one or more layers of weave knitted together. The roving can be quite heavy (16 ounces/square yard) or quite light (1/2 ounce per square yard), depending on the application.

Increasing core thickness increases strength.



Binder

The binder is just that. It holds all of the other materials together. It doesn't

add strength beyond holding the materials together. Early fiberglass products were built using a polyester resin. Polyester resin is cured with a catalyst — the more catalyst, the faster it cures. It is somewhat smelly and contains styrene, which can affect certain core materials. Vinylester resin is a stronger binder. It has a strength approaching that of epoxy and is cured with a catalyst. It still may contain styrene and sometimes has a unpleasant odor.

Epoxy is the strongest binder that I'll talk about. It uses a special hardener that must be mixed in a certain ratio. Adding more hardener to epoxy will not speed up its cure time and may prevent it from curing with the strength it was designed to have. Epoxy generally does not contain styrene or anything that would cause damage to most core materials.

There are various brands of all these binders. Most brands behave similarly to others. Comparing various data sheets may be the best way to determine which is best for your application.

Five minute and other quick setting epoxies are fine for quick repairs, but they aren't structural epoxies. The accelerators used generally don't allow the epoxy to achieve maximum hardness. They can be used for holding cores together before adding the fiberglass and for other handy, quick setting applications. If there is a small part that needs to be built, quick epoxies can be used and, since they set in so little time, fabricating parts is much quicker.

Core

For composite structures, the key is the core. The most common type is a foam core. There are many types of foam that can be used for a core; generally, they are all the closed cell types, rather than the white, bead board type. Common building styrofoam (usually blue or pink) is used in airplane wings and other places where shapes can be made by cutting the material with a hot wire. PVC foam is a very strong foam for its weight and is best for applications where flat sheets are used to form the structure.

Urethane foam is easy to carve and is used where complex shapes without a significant amount of strength are necessary. It is also useful for making molds.

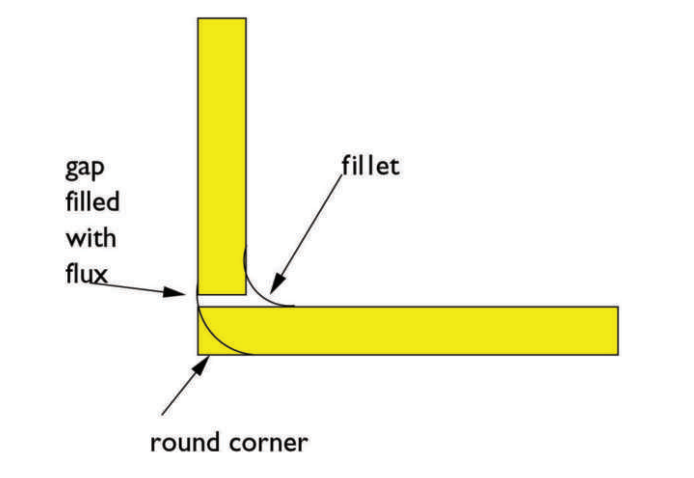
Sometimes, wood is used where extra strength is needed, like in the transom of a boat. Formula 1 race cars use an aluminum honeycomb core; this is very expensive and very strong.

The core acts as the webbing, just like in a steel I beam. The thicker it is — within reason — the stronger the part. This stiffness carries almost a 1:1 ratio. That is, if one part has a certain core thickness and another part has a core that is twice as thick, the stiffness will be approximately double for a thicker core. The core material will determine the amount of deflection that the part can withstand. If a part needs to withstand a crushing deflection, then a stiffer core will be required.

Some Handy Tools

Tools	**Uses**
Gloves/Hand Cream	Protects skin while working with resins
Scissors	Cut the fiberglass cloth
Utility knife	Cut the foam; trim edges of parts; cut finished parts for modification
Squeegees	Spread resin and slurry
Sandpaper	Finish edges of glass
Peel ply (polyester cloth)	Smooth edges; prep for additional layups; joints
Chip and acid brushes	Apply resin
Disposable cups (no wax)	Mixing and application of resin and slurry
Scale or pumps	Ratio resin for proper cure
Microballoons	Filler used to make slurry and fill holes
Flox	Cotton fibers used as a stronger filler
Polyethylene (poly)	Masking and for making tapes
Felt tip markers	Marking on poly, foam, and finished parts
Rotary tool	Trimming, drilling, grinding, various uses

A typical corner joint.



Fillers

Sometimes, it is necessary to give the binder something else to work with. There are various fillers that can be used for these applications. Generally, the fillers will reduce the strength of the binder, but their application can make the composition of the joint stronger or lighter to meet the application's needs.

The most common filler is microballoons, which are sometimes called micro. Microballoons are actually hollow glass beads that are very tiny, even microscopic. In handling them, you will find that they are somewhat of a cross between a dust and a liquid. Once stirred up, they tend to float. It is a good idea to wear some breathing protection when working with microballoons — getting a lung full probably is not something you want to try.

Microballoons are used to thicken or lighten a resin mix. When working horizontally or to fill a gap, the micro can be mixed to allow the resin to fill an area. A thin mix that is still quite runny is called a slurry. A medium mix — called wet micro — is a little runny. A very dry mix — called dry micro — is quite thick and will usually hold its shape until dry.

Another common filler is cotton flox, which is really chopped cotton strands, and is used as a filler when joining parts with a potential gap or fillet in a corner. Cotton flox is a little stronger than micro, but it is also heavier.

Other fillers are saw dust, wood flour, phenolic, cabosil, and talc. They can be used for various applications. Cabosil will thicken the resin both while it is uncured and cured.

Safety

Working with any of the various glass materials usually requires some safety precautions. Carbon fiber, being a natural material, can be a problem if you get a sliver. The body probably won't attempt to reject it, so it might be there for years. Slivers can be removed using common tweezers.

The binders (all of them) should be used with caution. Always wear eye protection — the catalyst for polyester resin can cause blindness. Even epoxy in the eyes is very uncomfortable and will probably require a trip to the hospital. Gloves or hand creams are a must. Among hand creams, Ply 9 is the best. For small layups, gloves can be used; disposable nitrile gloves provide better protection than latex gloves.

Soap and water are all that should be used if any resin gets on the skin, since other chemicals will force the resin deeper into the skin. Respirators may be required when using certain resins. Some people develop allergies to them, so care must be used when handling any resins. If a rash forms after using one, discontinue using it until the cause can be determined.

The foams can be cut with sharp knives. Never cut toward yourself and be aware of what is behind what you are cutting. Styrofoam can be cut with a hot wire (a bit of nichrome wire stretched in a bow), but urethane foam will emit noxious gases if cut with a hot wire.

A Common Layup

Working with resin is easiest when the temperatures are above 70° F. Most epoxies and other resins — unless designed for cold weather use — won't harden quickly enough at temperatures below 70° F.

Start with a flat sheet of foam — some 1/4" urethane, for example. Cut this into the proper shape with a utility knife. Using cloth cutting scissors, cut two pieces of bidirectional fiberglass. Cut it slightly oversized at a 45° bias with the roving at 45° relative to the edges of the cuts. Leave 1/4 to 1 inch extra around the edges of the part (this will be trimmed later). Once the glass is cut, roll it up and set it aside. Clean the foam with a vacuum to remove any dust or fiberglass particles.

You may notice that the glass is able to stretch or twist. This is normal; the weave isn't really tight and it allows some flexibility. If the weave is distorted too much, the strength will be lost. Rolling the glass is the safest way to keep the shape while moving the cut glass around.

Have a clean work station and cover it with plastic or

several layers of newspaper. Have all the tools handy with their covers loose. Put on your gloves, mask, and any other safety equipment before opening the epoxy.

Mix up some resin and hardener in the proper ratio using a scale or the pumps that are designed specifically for the resin. Only a small amount is needed at first. Resins should be mixed on disposable, wax-free paper or in plastic cups. Mix the resin thoroughly, making sure to get into the corners of the cup to be sure all of the resin and hardener are mixed.

Add some microballoons to the resin to make a slurry. The microballoons are tiny glass spheres that add little strength, but fill the resin to make it lighter. The slurry, when mixed, should be slightly runny when held up on the mixing stick. A consistency similar to loose toothpaste is perfect.

The slurry should be poured onto the foam. Use just enough to run a bead down the center of the foam. Using a squeegee, spread the slurry over the surface of the foam. The slurry will be used to fill the exposed cells of the foam; it will be strong enough to hold the glass to the foam without adding a significant amount of weight. Cover the entire surface of the foam with slurry, scraping any excess back into the mixing cup.

Before the slurry has had a chance to harden, roll out the first piece of fiberglass that was prepared earlier. Smooth this out as much as possible using your glove or cream-covered hands. If you are using a cream, allow enough time for the cream to dry, so it won't get on the glass.

Mix up another batch of resin. You may mix up one batch at the beginning, split some for the slurry, and use some here. This batch should be larger. As a rough estimate of how much you will need, mix about 10% more by weight than the weight of the glass used. This will be raw resin and hardener with no fillers.

Pour some of this resin on the glass. Using a squeegee, spread the resin around. The glass will become transparent and should have a dry look — not shiny and wet — with the weave still visible and no white spots. Wet out the glass to just past the edges of the foam — 1/8 to 1/4 inch. Look for air that may be trapped under the glass and try to work it out with a finger or the squeegee, being careful not to disturb the fibers. The fibers should all be laying flat and straight at a 45° angle.

Before that layer has hardened, roll out the second batch of glass on top of the last one. Pour some more resin on this. Use less than you did on the first layer, since there will usually be extra resin in that layer. Squeegee the resin out, keeping it even, not wet, and smooth. If there is extra resin, squeegee it to an edge and try to scoop it back into the cup. Extra epoxy doesn't add strength, only weight.

A handy help is to have some peel ply — a polyester cloth that doesn't stick well to epoxy. When it is applied to the edges and peeled off when the resin has cured, it will leave a rough area for easier bonding. We will use some peel ply on three edges now so that we can make a box from this later.

Roll out the peel ply over the wet glass. Using a brush, get the peel ply wet, if necessary. Trim the edges of the glass with scissors, limiting the overhang to about 1/4 to 1/2 inch, maximum. The weight of the untrimmed glass may cause a bubble near the edge of the part. This trim will help the glass adhere to the edges.

There are two resin bonds — chemical and mechanical. The chemical bond is made while the resin is wet. A mechanical bond is made after the resin has cured on a rough surface, where the wet resin can wrap itself around something. The peel ply will create a rough surface for mechanical bonding. Sanding the surface will help roughen it up. If peel ply is not available, sanding the cured part with 36 to 40 grit sandpaper creates a surface suitable for bonding.

Clean up the tools using paper towels, soap, and water. Dispose of any gloves and brushes. Extra resin will probably be wasted. Freezing epoxy can lengthen its life for up to a couple of days — just use a hair dryer to get it back to room temperature.

The resin will soon reach a stage called "green." This is when the glass is somewhat rubbery and flexible, but the fibers don't move. At this point, use the utility knife to flush trim the glass. It should cut easily. If the fibers move, wait a bit and try again. Depending on the resin and temperature, this could be one to 12 hours after the layup. If you miss the "green" stage, trimming can still be done with a little more effort and maybe a Dremel tool or some 36 grit sandpaper.

Once the layup has hardened, flip it over and repeat the process, putting two layers of glass on the back.

Using this same technique, you could build four more parts like this and attach them together using some fiberglass tape and flox. Remove the peel ply on the parts to be joined. Mix a small amount of resin and some flox to a toothpaste-like consistency. Mix up some raw resin and hardener.

Books

Composite Basics — Andrew C. Marshall

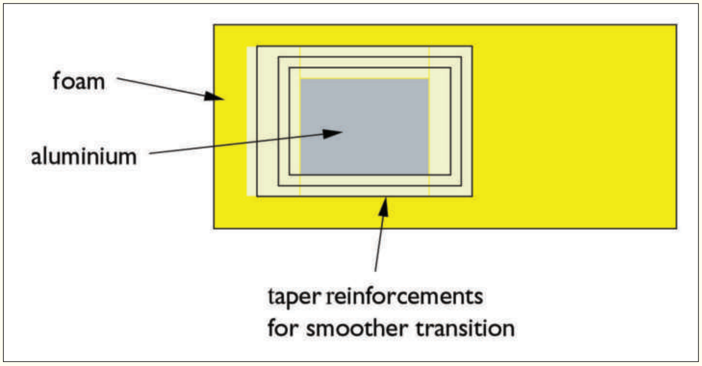
This book contains excellent information, including actual glass properties, strengths, and suggestions for high strength shapes.

Fiberglass Boat Building for Amateurs — Ken Hankinson

This one contains good information for large construction projects.

The Epoxy Book (**www.systemthree.com**)

This is a great reference oriented toward System Three products.



Adding points.

To make fiberglass tapes, mark some polyethylene with appropriately spaced parallel lines. Flip the poly over and lay some glass on top with the fibers at a 45° angle to the lines. Pour the mixed resin and hardener on the glass, then squeegee out the resin. The glass will start out opaque and then turn transparent. Fold the poly over the top and continue squeegeeing the excess to an edge. Cut the tapes on the lines drawn, leaving the poly on. The poly will help the tape hold its shape without frayed edges until it is in its final place.

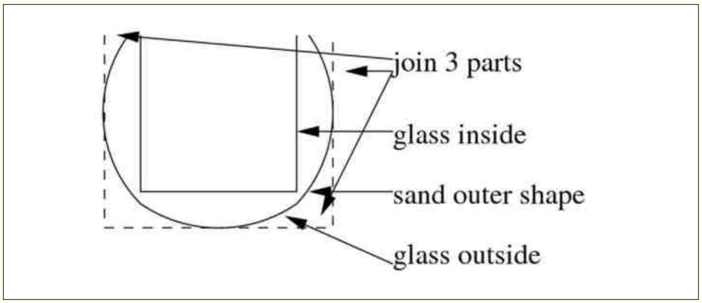
Put a little flox on the mixing stick and place some on the edge of the parts to be joined. Press the parts together, letting the flox ooze out the crack. Using your glove, cream-covered finger, or mixing stick, create a fillet in the joint.

Fiberglass won't bend like paper and there is a minimum bend radius that must be filled. Using a disposable chip or acid brush, brush on some raw, mixed resin about an inch in width on either side of the flox. Place the tape created earlier on the raw epoxy over the flox fillet, with the poly exposed. Press this tape in to smooth it out, then peel the poly off. Using some peel ply over the edges of the tape will smooth out the transition from the original layup to the tape. Clamp or brace these parts until the epoxy has hardened.

Curved Parts

Fiberglass is usually associated with curved things that look like they were formed by the wind. Our version of the wind is created by using an appropriate mold or formed core.

Hybid: round outside, square inside



A mold that is made of fiberglass or plaster is the most common type. For an occasional part, some foam that is covered with box sealing tape can be used. The plug or part to be copied is finished with a smooth, shiny finish. A mold release or wax is used on the inside of the mold to prevent the fiberglass from sticking to the mold. The fiberglass is laid over the mold and resin is worked into the glass, as it was for the flat parts above. Once the resin has cured, the part is carefully removed from the mold and trimmed as necessary.

If the inside of the piece can be somewhat square, a flat layup can be done on the inside of the part, leaving the outside foam bare. Some two inch (or other size, as appropriate) urethane foam can be laid up on the inside and joined. The outside of the parts can be carved to create the desired shape. Once the shape is finished, the outside can be glassed, either all at once or in sections. If you are going to do this in sections, peel ply the edges to be joined.

Hard-points

Sometimes, you might want to use bolts or other fasteners with the glass parts. Depending on the strength needed, multiple layers of glass can be used instead of the normal core material to provide a good place to put a bolt, pivot point, or other fastener. If pressed carefully, 33 layers of seven ounce, bidirectional fiberglass will come close to 1/4."

Embedding aluminum ingots of the same thickness as the core can provide localized hard-points. If the core is 1/4" thick or more, a small piece of aluminum can be embedded in the core in that spot. This aluminum can be tapped or drilled to create a hole for the fastener. The aluminum should be roughed up with sandpaper (40 to 36 grit) to allow a mechanical bond with the fiberglass to form. Again, be careful not to get the slurry on the metal.

The metal will need enough glass contact to pass the load on to the rest of the structure. Sometimes, tapered layups will be needed locally around the hard-point. Start with something one or two inches larger than the hard-point. Subsequent layups can be 1/4 to 1/2 inch less on all sides.

Finishing

For indoor robots, any finish would be for aesthetics only. Almost any paint suitable for plastics should work fine. Rough up the glass before applying any paints or glues to insure a good mechanical bond.

If the robot is to be used outdoors, the resin must be protected. Ultraviolet radiation will break down the resin quite quickly, often in just a few months. There are several UV-blocking primers available. I usually use Smooth Prime,

since it is also a filler and removes one step from the finishing process. Some gel coat finishes are also acceptable.

Outdoor projects may absorb a great deal of heat. If a project is painted black and put outside on a sunny, 90° F day, the surface temperature can reach 175° F. Depending on the core material, this can cause structural damage or soften the epoxy. Composite structures should be painted a light color. White is the most common color, but yellow or silver will keep the temperatures manageable.

Filling the weave with microballoons will give the robot a fresh-out-of-the-mold look. A thin coat (less than 1/16") of dry micro will fill the weave; sand this smooth with 80 grit sandpaper. Leave only enough micro to fill the weave; you should still see the high spots of the glass. Apply a coat of good filler/primer to fill the sanding scratches and you will have a good prep on the robot for your favorite paint.

I used to hate sanding, but, since I have been working with composites, I love it. Like anything, using the proper tools makes all the difference. Get 36 or 40 grit sandpaper to rough up the glass for finishing the edges and use a new piece when one wears out. Try using a small sanding block — it really makes a difference. Use 80 grit sandpaper on the micro for the nicest blends and smoothest finishes. A 220 grit piece is ideal after finishing the primer to prep for the final paint coat. Using any other grits or worn paper will make the job harder.

Common Problems

I have run into most of these problems at one time or another and they generally have little effect on the finished part if they are dealt with properly.

• **Waxy film on the surface of the parts.** This is amine blush, which is usually caused when the resin is curing and the humidity is high. It washes off with hot water and soap or can be sanded off. Some resins are more susceptible to amine blush than others. If you get it on every layup, try switching resins.

• **The resin cured in the cup before I was done.** The resin cures using self-generated heat (exotherm). In large quantities, the process is accelerated. Avoid large quantities in hot environments or spread out the resin as soon as it is mixed. If you are using a catalyst for curing resins, use less catalyst to prevent fast curing.

• **The resin isn't hardening.** Check the temperature. If it was cold overnight, wait another day or use a small heater (carefully!) to accelerate hardening. Make sure you used the correct ratio; adding too much hardener to epoxy will prevent it from hardening, no matter what you do.

• **I need to join parts where there is no peel ply.** When I run out of or forget peel ply, it isn't a big deal. Carefully sand the surface of the parts, trying not to damage the glass fibers. Sand enough to create a rough surface where the epoxy can form a mechanical bond.

• **It is easy to pull the tapes away.** If the edges of the tapes or any joints aren't peel plied, they can leave places where something can get underneath and cause them to pull away. It is a good idea to peel ply all edges. If the parts weren't prepped properly (sanded or peel plied), the mechanical bond may not be able to form.

• **I am building a big part and the glass isn't big enough to cover it in one piece.** If you are using bid cloth, overlap the edges of the two parts by about an inch. Overlap less if you are using a thin cloth and more if you are using something heavy. While the glass is wet, peel ply this edge. If you are using uni cloth and the glass isn't wide enough,

Resources

Most of the epoxy or polyester resins — along with glass and other supplies — can be purchased at a good marine supply store.

The System Three Corporation has a trial kit for $20.00. It includes resin, fillers, glass, tools, and everything else needed to try out composite construction (www.systemthree.com). Their free book on epoxies is a great resource.

Wicks Aircraft (www.wicksaircraft.com) has a huge variety of foam, wood, and aluminum core materials. They also have epoxy, vinylester, and polyester resins, along with glass and other supplies.

butt the parallel fibers against each other.

• **I forgot to wash my tools and now they don't work.** Squeegees can sometimes be flexed to make the hardened resin flake off. Scissors will need to be taken apart if the epoxy got into the hinge. A normal scissors sharpener will clean the blades. A wire brush or fine sandpaper can clean the hinge. Clothes will crack where epoxy hardens on them.

• **I finished the part, but over-stressed it. Now it has a crack**. First, be sure the part was designed appropriately for the load. If a fragile part was accidentally damaged, it can be repaired easily. Cut out the crack, exposing the core. Repair or fill it as necessary, even if this requires removing the original core and replacing it to maintain a specific weight. You can use a slurry to "glue" the replacement core in. Sand the edges out at least an inch past the hole and replace the glass, using the same number of layers as the original part had. Taper the sanded edge inward slightly and peel ply the edges.

• **The part is full of bubbles between the layers.** This is fairly common. If there aren't too many bubbles, nothing will need to be done. You'll have to decide what the safety factor of your part is; 10-20% of the area is probably okay. If the bubble is localized, a syringe filled with pure resin can be injected into the bubble to add weight and some strength. If there are too many bubbles, just discard the part or use it for a test subject.

• **The part looks freckled, like it has white spots all over.** You need to use more resin; these white spots are the places where the resin is too sparse.

• **My part is really shiny.** There is too much resin on the part. If it is cold out, the tendency is to add more resin until the part is completely wet. The cold thickens the resin, so it doesn't wet out as fast. This extra resin only adds weight, not strength. Use a hair dryer to keep the parts and the resin warm if the day is a little colder than desired.

• **My part isn't as strong as I would like. Can I just paint on epoxy to strengthen it?** No! The resin is a binder. Most of the strength comes from the glass, but there is a synergy. The strength of the composition is more than that of the individual parts. Putting on another layer of glass and resin will probably add strength, but take another look at your design. It might be time to start over.

Conclusion

Building with composites offers almost unlimited flexibility for designing your robot. The finished design can take almost any shape and strength. Changes are quite easy to make and work is mostly done with simple hand tools. With practice, composite structures can be built quickly and reliably. SV

Robotics Resources

The Robot Builder's Bookshelf

Even in this age of the Internet, books are still the premier tool for learning about a subject. They're relatively cheap and the better ones pack more information than you'll find on a galaxy of web pages. I say this not only as a book author myself, but as a voracious book reader. At one time or another, I've probably owned every book on robotics that has ever been published.

by Gordon McComb

Gordon McComb
The Robot Builder's Bookshelf
T&L Publications

Here are some books for the robot-minded that you'll want to consider adding to your reading list. They are available for purchase — new or used — but don't forget about your local library. If a book you want isn't carried by your library, ask the librarian if it is available from another branch or ask that the book be considered for purchase.

Of course, the following list doesn't include every robotics book. There are just too many, but these are among the most relevant to amateur robotics and its allied fields.

Online or Local

Even some highly technical books are usually carried by the local bookstores. If you have a good bookstore nearby, check out the Science & Technology (or similar) section for books on robotics, electronics, and related topics.

Odds are, though, many of the robot books you'll want to browse or buy are not available locally, except through special order. Another option is to order them online.

First, consider the *SERVO* and *Nuts & Volts* Hobbyist Bookstores. The books have been carefully selected by the magazine editors for content and value. Check out this month's ad for the Hobbyist Bookstore or go online to **www.nutsvolts.com** or **www.servomagazine.com** and click on the Online Store or Books link.

To make things even easier, click on the Robotics Resources Recommends link for the available titles featured here.

Additional sources for online books are:

Amazon Books
www.amazon.com

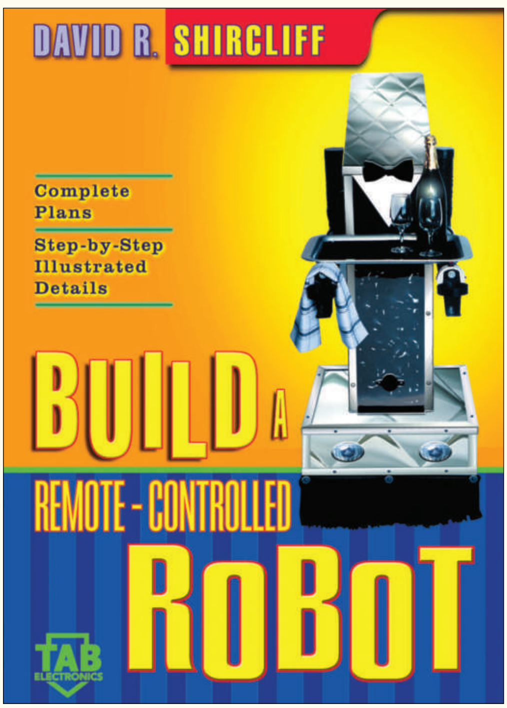
Best-known of the online bookstores. Books can be ranked by publication date (great for seeing what's new), bestsellers, author, title, and customer reviews.

Barnes and Noble
www.barnesandnoble.com

The online subsidiary of the Barnes & Noble bookstore chain. Not as many robotics titles as Amazon, but the ones they have are likely to be the ones you want.

Lindsay's Technical Books
www.lindsaybks.com

Old (reprints) and new books on electronics, how-to (woodworking, metalworking, etc.), high voltage, and other unusual topics. Lindsay sells a large assortment of metal-working and foundry books — some old and some new — and books on plastic injection molding and plastic vacuum forming.



OpAmp Technical Books
opampbooks.com

Online and local (Los Angeles, CA) retailer of technical books. Limited selection of robotics books, but offers to beat Amazon's price.

Powell's
www.powells.com

Offers a fair number of robotics titles. Displays the list in alphabetical order, by author, or by price. Occasionally offers discounts. Local stores in Oregon.

Robot Books
www.robotbooks.com

A reseller of books, toys, movies, and other robotic artifacts. For robotics and other books, they connect with Amazon.

San Diego Technical Books
www.booksmatter.com

Online and local (San Diego, CA) book retailer. Good selection and you have the ability to list books by publication date, author, title, or selling rank.

Square One Electronics
www.sq-1.com

Publishers of how-to books on PIC microcontroller programming and stepper motors. Available from the company or from many online booksellers, such as Amazon. Titles include: *Easy PIC'n*, *PIC'n Up the Pace*, *PIC'n Techniques*, and *Serial PIC'n*.

Workshop Publishing
www.buildstuff.com

Plans include: Table Top Vacuum Forming Machine, Secrets of Building a Vacuum Forming Machine, and Secrets of Building an Injection Molding Machine. Books include: *Vacuum Forming for the Hobbyist*, *Understanding Thermoforming*, *Mold Making and Casting Guides*, and *How to Cast your own Plastic Parts*.

Books on Amateur Robotics

Consider these books on amateur robotics. Some are a few years old, but still offer a good deal of relevant information.

Note that I provide the name of the author, publisher, publication date, and ISBN. If you're interested in ordering a book online or through your bookstore, all you really need is the ISBN and the title to verify that you've gotten the right one.

123 Robotics Experiments for the Evil Genius
Myke Predko
McGraw-Hill, 2003

ISBN: 0071413588

This large format book is presented like a lab journal (but a fun one!), with projects of one or two pages — complete with parts needed and all the steps.

Absolute Beginner's Guide to Building Robots
Gareth Branwyn
Pearson Publications, 2003
ISBN: 0789729717

A quite engaging book that teaches the elementals of robot building while practicing with three "junkbots" built from stuff around the house. The author's easy going style makes this book a fun read.

Applied Robotics
Edwin Wise
Delmar Learning, 1999
ISBN: 0790611848

Grab-bag of amateur robotics, with a strong emphasis on hardware design (using an Atmel AVR microcontroller). Keynote: included on the CD with book is "Fuzbol" — developed by the author's company — a fuzzy-logic robotics operating system. A second edition is now available.

Build a Remote Controlled Robot
David R. Shircliff
Tab Books, 2002
ISBN: 0071385436

Construction plans for a remote controlled robot vehicle.

Build Your Own Combat Robot
Pete Miles, Tom Carroll
McGraw-Hill, 2002
ISBN: 0072194642

Building battling robots — like the ones on *BattleBots* and *Robotica*. Provides handy construction tips for heavy-duty robots, even if you don't want to attach lawn mower blades to the thing.

Build Your Own Robot!
Karl Lunt
A K Peters, Ltd., 2000
ISBN: 1568811020

Reprint of Karl's columns from *Nuts & Volts Magazine*; the columns appeared in the early to mid 1990s and are still very useful. Portions of the book get *very* technical, so don't try this one if you're a beginner.

Insectronics: Build Your Own Walking Robot
Karl Williams
McGraw-Hill, 2002
ISBN: 0071412417

An intriguing selection of robot projects based around insect-like designs. Most of the constructions are of aluminum. The author also offers a companion volume, *Amphibionics: Build Your Own Biologically Inspired Reptilian Robot* (ISBN: 007141245X) and his new book, *Build Your Own Humanoid Robots* (ISBN: 0071422749).

Introduction to Robotics, An
Harprit S. Sandhu
Nexus Special Interest, Ltd., 1997
ISBN: 1854861530

Nice entry-level guide to robotics, including the construction of a servo-operated robot.

Muscle Wires Project Book
Roger G. Gilbertson
Mondo-Tronics, 1993
ISBN: 1879896141

Book and kit about "Muscle Wires," a brand of shape memory alloy available from the author's company (Mondo-tronics). Projects include Boris, a walking robot.

Personal Robot Navigator, The
Miller, Winkless, Phelps, Bosworth
Robot Publishing, 1999
ISBN: 188819300X

Mostly theory, this book also contains a software CD of a robot navigation simulator (Robonav).

PIC Robotics: A Beginner's Guide to Robotics Projects Using the PICMicro
John Iovine
McGraw-Hill, 2002
ISBN: 0071373241

Robot project book using the PIC microcontroller as the central brain.

Practical Robotics: Principles and Applications
Bill Davies
CPIC Technical Books, 1997
ISBN: 096818300X

General book on different mechanical construction ideas, with practical details. There's not much actual robot building in this one, but the pieces can readily be put together to make one.

Robot Builder's Bonanza, Second Edition
Gordon McComb
McGraw-Hill, 2000
ISBN: 0071362967

My book on beginning and intermediate robotics. I'm proud to say it's the bestselling book on amateur robotics; the first edition came out in 1987.

Robot Builder's Sourcebook
Gordon McComb
McGraw-Hill, 2002
ISBN: 0071406859

Over 700 telephone book-sized pages on robotic parts, from what it is to where to buy it. Includes numerous, not-so-obvious sources and materials ideas.

Robot Building for Beginners
David Cook
APress, 2002
ISBN: 1893115445

For raw beginners with little or no experience. Good for those just starting out. Another book by this author, *Intermediate Robot Building* (ISBN: 1590593731), is due out shortly.

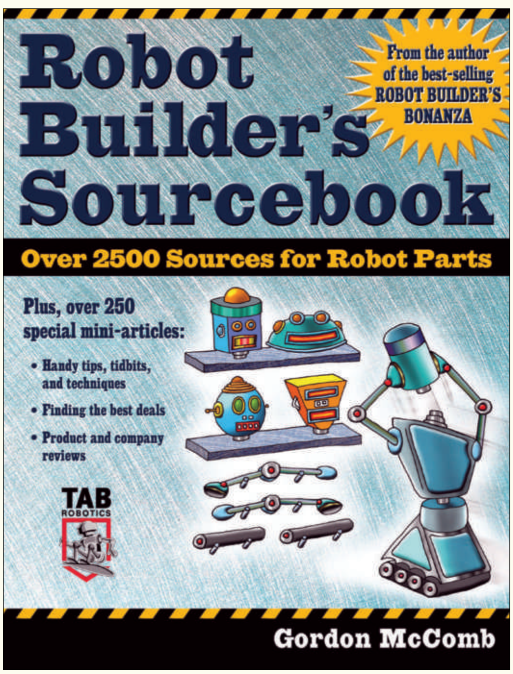
Robot Building for Dummies
Roger Arrick, Nancy Stevenson
Wiley, 2003
ISBN: 0764540696

Roger runs the **Robots.net** site and operates his own robotics manufacturing company. This books actually goes beyond entry level and includes some intermediate topics.

Robot DNA (series)
Various authors
McGraw-Hill, 2002-2004

A unique series of books on robotics construction that I co-edited with Myke Predko.

Each book is aimed at the intermediate to advanced roboteer and details a specific aspect of construction. Books in the series — so far — consist of *Building Robot Drive Trains* (ISBN: 0071408509), *Programming Microcontrollers* (ISBN: 0071408517), and my own *Constructing Robot Bases* (ISBN: 0071408525).



Robots, Androids, and Animatrons, Second Edition
John Iovine
McGraw-Hill, 2001
ISBN: 0071376836

Several entry level projects, including a small, six-legged walking robot.

Stiquito for Beginners: An Introduction to Robotics
James Conrad, Jonathan Mills
IEEE Computer Society Press, 1999
ISBN: 0818675144
and
Stiquito: Advanced Experiments with a Simple and Inexpensive Robot
James Conrad, Jonathan Mills
Institute of Electrical and Electronic Engineers, 1997
ISBN: 0818674083

Beginner and advanced books on constructing small, legged robots using shape memory alloy wire (e.g., BioMetal, Nitinol, Dynalloy, or Muscle Wires). Both books list the basic parts you'll need to build the sample robot.

LEGO Robotics and LEGO Building

LEGO Mindstorms continues to be one of the best ways to get into the field of robotics. These books will help you get beyond what comes in the box.

Building Robots with LEGO Mindstorms
M. Ferrari, G. Ferrari, R. Hempel
Syngress Media, Inc., 2001
ISBN: 1928994679

Highly recommended guide to intermediate and advanced level LEGO Mindstorms robotics. Lots of mechanics; many programming examples are in Not Quite C (NQC).

Creative Projects with LEGO Mindstorms
Benjamin Erwin
Pearson Education, 2001
ISBN: 0201708957

Using color illustrations, this book demonstrates over a dozen fun and unusual LEGO Mindstorms creations. A great book for the classroom.

Dave Baum's Definitive Guide to LEGO Mindstorms
Dave Baum
APress, 1999
ISBN: 1893115097

Part of this book is on LEGO Mindstorms mechanics and part is on using the author's NQC programming language — which he wrote especially for the Mindstorms platform. The book is now in its second edition.

Extreme Mindstorms: an Advanced Guide to LEGO Mindstorms
Baum, Gasperi, Hempel
APress, 2000
ISBN: 1893115844

Collection of intermediate level topics on LEGO Mindstorms. The chapters are written by a variety of well-known Mindstorms experts.

Jin Sato's LEGO Mindstorms: The Master's Technique
Jin Sato
No Starch Press, 2002
ISBN: 1886411565

Advanced level LEGO Mindstorms robots, such as a robotic dog. See also *Joe Nagata's LEGO Mindstorms Idea Book* (ISBN: 1886411409) from the same publisher.

Technical Robotics, Theory and Design

These books are among the many technical tomes and textbooks on robotics and robotics theory.

Computational Principles of Mobile Robotics
Gregory Dudek, Michael Jenkin
Cambridge University Press, 2000
ISBN: 0521560217

Math-heavy with an emphasis on computation and algorithms; intended for advanced undergraduate and graduate students studying mobile robotics.

Mobile Robots: Inspiration to Implementation
Jones, Flynn, Seiger
A K Peters, Ltd., 1998
ISBN: 1568810970

Hands-on guidebook to constructing mobile robots. Offers two main projects: Tutebot, which introduces basic mobile robot concepts, and Rug Warrior, based on the Motorola MC68HC11 microcontroller.

Robot Evolution: The Development of Anthrobotics
Mark Rosheim
Interscience, 1994
ISBN: 0471026220

Technical overview on robots and robot systems for the hardcore.

Robotic Explorations: An Introduction to Engineering Through Design
Fred Martin
Pearson Education, 2002
ISBN: 0130895687

This is an expensive book, but it's one of the best for teaching the fundamentals of robotics in the classroom. Written by a respected robotics professor, it is a textbook suitable for first or second year robotics courses in high school and college. The author provides explicit examples and exercises using LEGO Technic pieces and the MIT Handy Board controller, but you can apply what you learn to almost any robotics platform.

Sensors for Mobile Robots: Theory and Application
H. R. Everett
A K Peters, Ltd., 1995
ISBN: 1568810482

Robots are nothing without sensors that allow them to see. This is an overview book designed to help familiarize you in the role of sensors in robotics and the available technologies. Written by one of the experts in the field.

Artificial Intelligence and Behavior-Based Robotics

The following books detail various approaches to endowing your robotic creations with artificial intelligence. Among the most popular AI methods today is behavioral programming, which uses one of several approaches.

Artificial Intelligence and Mobile Robots
Kortenkamp, Bonasso, Murphy
MIT Press, 1998
ISBN: 0262611376

A review of functional systems and techniques in artificial robots used by researchers and companies.

Behavior-Based Robotics
Ronald C. Arkin
MIT Press, 1998
ISBN: 0262011654

Overview text of various behavior-based robotics techniques. The book contains references to robots that have used the various AI techniques described, with moderate detail into their implementation.

Cambrian Intelligence: The Early History of the New AI
Rodney Allen Brooks
MIT Press, 1999
ISBN: 0262024683

A collection of MIT professor Rodney Brooks' earlier works on robotics intelligence.

Introduction to AI Robotics, An
Robin R. Murphy
MIT Press, 2000
ISBN: 0262133830

A round-up of several popular approaches to artificial intelligence in robotics. Fairly technical, but still readable.

Mobile Robotics: A Practical Introduction
Ulrich Nehmzow
Springer-Verlag, 2003
ISBN: 1852337265

An introductory textbook aimed at the college student studying mechatronics or robotics. Emphasis is on mobile robots and is heavy on concepts and theory, not actual construction.

Robot: Mere Machine to Transcendent Mind
Hans Moravec
Oxford University Press, 2000
ISBN: 0195136306

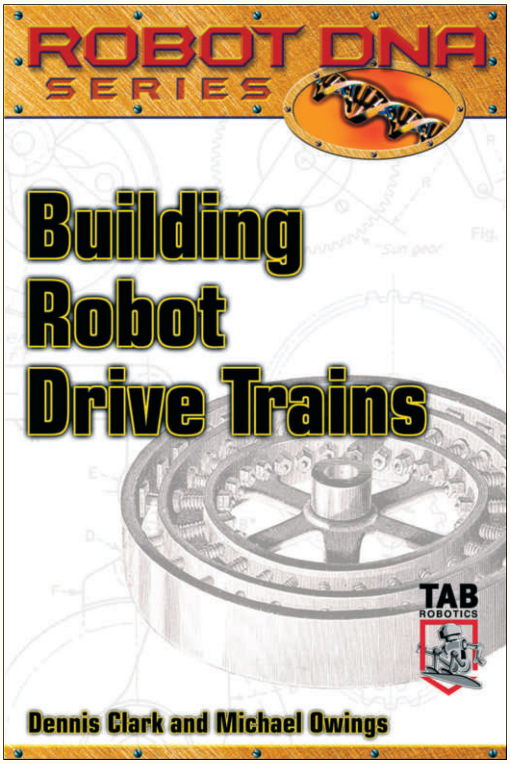
A speculative — but informed — look at one possible future with intelligent robots written by a professor at Carnegie-Mellon University. See also the author's *Mind Children: The Future of Robot and Human Intelligence* (ISBN: 0674576187).

Robot Programming: A Practical Guide to Behavior-Based Robotics
Joe Jones, Daniel Roth
McGraw-Hill, 2003
ISBN: 0071427783

Robotics mavens provide practical examples of implementing behavioral programming.

Vehicles: Experiments in Synthetic Psychology
Valentino Braitenberg
MIT Press, 1986
ISBN: 0262521121

A small — but highly influential —



book on how simple machines can mimic living organisms.

Mechanical Design

Robots are a mixture of systems: mechanics, electronics, and programs. The following books are useful for honing your construction skills in the shop. They are general interest and not specifically aimed at robot building.

Beginner's Guide to Reading Schematics
R. J. Traister, A. L. Lisk
Lindsay Publications, 1984
ISBN: 0917914252

Short, but sweet, this book covers the foundations of basic mechanics, such as the gear, pulley, and windlass. The mechanisms are shown in self-explanatory illustrations. The book is a reprint from the late 19th century; I have a version of the book published in the 1950s.

Home Machinist's Handbook
Doug Briney
McGraw-Hill, 1984
ISBN: 0830615733

Covering basic machine shop practice for the home, this perennial bestseller is the ideal companion for the intermediate or advanced robot builder. The emphasis is on machining in the home "lab" and the example tools are common Sherline desktop mills and lathes.

Illustrated Sourcebook of Mechanical Components
Robert O. Parmley (Editor)
McGraw-Hill, 2000
ISBN: 0070486174

Gargantuan book on mechanical design and components. The typical and atypical are covered.

Mechanical Devices for the Electronics Experimenter
Britt Rorabaugh
Tab Books, 1995
ISBN: 0070535477

Overview of mechanical construction of common components, such as motors and linkages.

Mechanisms and Mechanical Devices Sourcebook
Neil Sclater, Nicholas P. Chironis
McGraw-Hill, 2001
ISBN: 0071361693

Over 2,500 mechanical device elements and 1,200 illustrations. Never wonder how something works; look it up in this terrific resource. Special chapters on robotic systems, couplings and linkages, gears, and pneumatics.

The Prop Builder's Molding & Casting Handbook
Thurston James
F & Q Publications, 1989
ISBN: 1558701281

You don't have to be a Hollywood movie prop builder to enjoy this book, though it's really written for any kind of prop builder. Various materials and finishing techniques are covered; the molding and casting techniques are useful in robotics applications, such as making "sticky" tires for sumo robots.

Tabletop Machining
Joe Martin, Craig Libuse (Illustrator)
Sherline Products, Inc., 1998
ISBN: 0966543300

Written by the head of Sherline — the desktop machine manufacturer — this handy tome teaches you the basics of using small lathes and mills. The book is most suitable for owners of Sherline products — as you would imagine — but is applicable to almost any other brand.

Electronics How-to and Theory

General interest books about electronics.

Art of Electronics, The
Paul Horowitz, Winfield Hill
Cambridge University Press, 1990
ISBN: 0521370957
and
Student Manual for the Art of Electronics
Paul Horowitz, T. Hayes
Cambridge University Press, 1990
ISBN: 0521377099

The Art of Electronics and its companion student workbook are the standard texts for electronics in many schools. Fairly laden with theory and math, the books are, nevertheless, aimed at a general academic and hobbyist audience.

Bebop to the Boolean Boogie
Clive Max Maxfield, Pete Waddell
Butterworth-Heinemann, 2002
ISBN: 0750675438

Unconventional approach (including humorous illustrations) to teaching logic circuits. Readable and quite engrossing. An extensive index makes it easy to locate important topics.

Beginner's Guide to Reading Schematics
R. J. Traister, A. L. Lisk
Tab Books, 1991
ISBN: 0830676325

Like the title says: how to read schematics, for beginners.

Build Your Own Low Cost Data Acquisition and Display Devices
Jeffrey Hirst Johnson
Tab Books, 1993
ISBN: 0830643486

Specializes in data acquisition on PC compatible computers.

CMOS Cookbook
Don Lancaster
Elsevier, 1998
ISBN: 0750699434

One of two desk references no electronics hobbyist or engineer should be without. The other is *TTL Cookbook* (ISBN: 0672210355), also by Don Lancaster. You can read much more by Don at his website: **www.tinaja.com**

Electronic Circuit Guidebook
(various volumes)
Joseph J. Carr, Natalie Harris
Delmar Learning
Volume 1: *Sensors*; 1997,
ISBN: 0790610981
Volume 2: *IC Timers*; 1997,
ISBN: 0790611066
Volume 3: *Op Amps*; 1997,

ISBN: 0790611317
Volume 4: *Electro Optics*; 1997,
ISBN: 0790611325
Volume 5: *Digital Electronics*; 1998;
ISBN: 0790611295

Reference books on various electronics topics of interest to all engineers.

Forrest Mims Engineer's Notebook, The

Forrest M. Mims, Harry L. Helms
Elsevier, 1993
ISBN: 1878707035

Most anyone involved in electronics knows Forrest Mims — a prolific writer who has authored numerous magazine articles, columns, and books since the 1960s.

This book is a compilation of how-to and circuit design guidance that is suitable for all students of electronics. See also, from the same author, *The Forrest Mims Circuit Scrapbook* (ISBN: 1878707493).

IC Op-Amp Cookbook

Walter G. Jung
Pearson Education, 1997
ISBN: 0138896011

Various ways op-amps can be used in circuits. More of a reference.

Logicworks 4: Interactive Circuit Design Software for Windows and Macintosh

Capilano Computing Systems (author)
Prentice-Hall, 1998
ISBN: 0201326825

Book and CD-ROM with a student version of the LogicWorks circuit design and simulation software.

Making Printed Circuit Boards

Jan Axelson
McGraw-Hill, 1993
ISBN: 0830639519

Covers various techniques for constructing prototype circuit boards.

Practical Electronics for Inventors

Paul Scherz
Tab Books, 2000
ISBN: 0070580782

A unique book with a unique angle, this text demonstrates the how and why of electronics circuits for people who may not want to learn all the nitty-gritty details. The book still has some technical formulas, but the emphasis (like the title says) is on practical electronics.

Printed Circuit Board Materials Handbook

Martin W. Jawitz (editor)
McGraw Hill, 1997
ISBN: 0070324883

Intended for the electronics professional, this book covers various materials and techniques used in printed circuit board manufacture.

Teach Yourself Electricity and Electronics

Stan Gibilisco
Tab Books, 2001
ISBN: 0071377301

Introduction to electronics, with self test exercises.

Interfacing to PC

Consult these books if you're interested in building a robot tethered to a PC or based on PC architecture.

Parallel Port Complete

Jan Axelson
Lakeview Research, 1997
ISBN: 0965081915

Everything you ever wanted to know about interfacing with the parallel port on your PC compatible. Jan is arguably the leading author on port interfacing.

See also, from the same author and publisher: *Serial Port Complete* (ISBN: 0965081923) and *USB Complete* (ISBN: 0965081958).

PC PhD: Inside PC Interfacing

Myke Predko
Tab Books, 1999
ISBN: 0071341862

Hardware and software for interfacing to a PC, including serial, parallel, and bus-level boards.

Programming the Parallel Port

Dhananjay V. Gadre
CMP Books, 1998
ISBN: 0879305134

Using the parallel port of the PC compatible computer to control relays and other devices and to accept input from sensors.

Program examples in the C language.

Real World Interfacing with Your PC

James Barbarello
Delmar Learning, 1997
ISBN: 0790611457

Entry level guide on using the serial and parallel ports of the PC compatible to interface with various things.

Use of a PC Printer Port for Control & Data Acquisition

Peter H. Anderson
Self-published, 1996
ISBN: 0965335704
and

Parallel Port Manual Volume 2, The

Peter H. Anderson
ISBN: 0965335755

Fairly in-depth coverage of parallel port interfacing on the PC. They contain a number of unique topics not found in similar books, such as using an ultrasonic sensor and interfacing to a compass module. Most of the projects are useful in robotics. **SV**

About the Author

Gordon McComb is the author of the bestselling *Robot Builder's Bonanza*, *Robot Builder's Sourcebook*, and *Constructing Robot Bases*, all from Tab/McGraw-Hill. In addition to writing books, he operates a small manufacturing company dedicated to low cost amateur robotics. You're welcome to visit at **www.budgetrobotics.com** He can also be reached at robots@robotoid.com

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

May and June bring several big international robot events — including the Western Canadian Robot Games, the Singapore Robotic Games, Eurobot, and RoboCup, which is in Portugal this year. There are plenty of events planned in the US, too, so you should be able to find something interesting to attend, no matter where you live.

I've received many queries lately as to whether or not there will be another DARPA Grand Challenge. All I can say — so far — is that DARPA has confirmed that there will be another event, but it will be six to eight weeks before any details are announced. My guess is that it will be scheduled for 2006. Check upcoming issues of SERVO and the Robots.net website for updates.

— R. Steven Rainwater

For last minute updates and changes, you can always find the most recent version of the complete Robot Competition FAQ at Robots.net:
http://robots.net/rcfaq.html

May

2 **Swiss RobOlympics**
University Rapperswil, Switzerland
Line following, mini sumo, and other competitions for autonomous LEGO Mindstorms robots.
http://www.robolympics.ch/

5 **Micro-Rato**
University of Aveiro, Portugal
You might know it better as micro-mouse. Mice, rats, some kind of robotic rodents ...
http://microrato.ua.pt/

6-16 **Edventures Robotics Challenge**
Online event
This is a rather unique event. The challenge is not against other robot builders. Each team attempts to complete the challenge and provides digital photos to the judges, who make a final determination. There is not a single winner, in the traditional sense.
www.edventures.com/erc/

14 **SPURT**
Rostock-Warnemunde, Germany
SPURT stands for School Projects Using Robot Techniques. Special autonomous SPURT racing robots race around a special SPURT track.
http://spurt.uni-rostock.de/

15 **Atlanta Robot Rally**
SciTrek Museum, Atlanta, GA
The annual Robot Rally includes a vacuuming contest, mini-sumo, and a unique open contest in which contestants provide their own goals.
www.botlanta.org/Rally/

15 **14th Western Canadian Robot Games**
Southern Alberta Institute of Technology, Alberta, Canada
Whether you bring a bot or build one in workshops at the event, it can compete for prizes in different levels of sumo, line following, mine sweeping, LEGO Mission Mars, media challenge, and more.
www.robotgames.com

18-20 **Singapore Robotic Games**
Singapore Science Centre, Republic of Singapore
This one includes a wide range of events, from micromouse and sumo to pole balancing and wall climbing.
http://guppy.mpe.nus.edu.sg/srg/

21-23 **Eurobot**
La Ferte Bernard, France
Each year, a different event is offered. This year, it's robot rugby.
www.anstj.org/robot/concours/eurobot/garde_en.html

26 **NATCAR**
National Semiconductor HQ, Santa Clara, CA
University teams compete to build the best autonomous race car that uses optical or magnetic line following sensors to stay on the track. See the UC team sites for information.
www.ece.ucdavis.edu/natcar/
www.ee.ucla.edu/natcar/

28-31 **BattleBots**
Mare Island (Vallejo), CA
If you're into watching radio controlled vehicles

bump into each other, this is the big one.
www.battlebotsiq.com

29-30 Robot Assault
Huntsville, AL
More radio controlled vehicle destruction.
www.secr.org

June

4-6 RoboJoust
Las Vegas, NV
Another radio controlled vehicle demolition derby.
www.robojoust.com

5 Argentine Championship Robot Soccer
Buenos Aires, Argentina
Middle League SimuroSot robot soccer.
http://www.exa.unicen.edu.ar/cafr2004/

6 PDXBOT
Smith Center Ballroom, Portland State University, Portland, OR
Lots of events, including mini-sumo, micro-sumo, Japan sumo, line following, walkers, and a talent show.
www.portlandrobotics.org/PDXBOT/

12-14 AUVS International Ground Robotics Competition
Oakland University, Rochester, MI
An autonomous ground vehicle must navigate an outdoor obstacle course within a prescribed amount of time, while staying within a five mph speed limit.
www.igvc.org/deploy/

19 UK National Micromouse Competition
Technology Innovations Center, Birmingham, UK
Robot mice strive to win the coveted brass cheese.
www.tic.ac.uk/micromouse/

25-27 MATE ROV Competition
University of California, Santa Barbara, CA
Student built ROVs must locate and retrieve objects of varying sizes and shapes.
www.marinetech.org/rov_competition/

27-7/4 RoboCup Robot Soccer World Cup
Lisbon, Portugal
This event includes the Soccer Simulation League, Small-Sized Robots, Mid-Sized Robots, Four Legged (AIBO) Robots, and RoboCup Junior. The goal is to field a team of robots that can beat the world's best human team by 2050.
www.robocup.org

July

17 DPRG Table Top Robot Contest
The Science Place, Dallas, TX
Mini-robots compete on table top sized courses in line following, sumo, and other events.
www.dprg.org/competitions/

19-23 K'Nex K*bot World Championships
Las Vegas, NV
Three events are included: two-wheel drive autonomous K*bots, four-wheel drive autonomous K*bots, and radio controlled Cyber K*bots.
www.livingjungle.com/

25-29 AAAI Mobile Robot Competition
San Jose Convention Center, San Jose, CA
Robots must navigate the conference center in the "Robot Challenge." They must locate injured humans in "Robot Rescue" and serve humans in "Hors d'oeuvres, Anyone?"
www.aaai.org/Conferences/National/

25-29 Botball National Tournament
San Jose, CA
Held in conjunction with the National Conference on Educational Robotics and timed to coincide with this year's AAAI convention.
http://www.botball.org/

28-Aug 1 AUVS International Aerial Robotics Competition
US Navy TRANSDEC, San Diego, CA
Autonomous underwater robots must locate a target at the bottom of the test arena, mark it, and proceed to a recovery zone before surfacing.
http://www.auvsi.org/competitions/water.cfm

Down Memory Lane

by Gerard Fonte

When Dan asked me to write an "Appetizer" column, I was surprised and flattered. While I write an engineering mentoring column for *Nuts & Volts* (the sister magazine to *SERVO)*, I had never seen this publication in its present form. So, I went to the bookstore and got a copy. Building robots reminded me of all the things I learned out of high school, back around 10 BC (Before Computers). Since the statute of limitations has expired, I am able to discuss some of the more colorful attributes of typical nerd behavior for the time.

I found outside, hands-on learning to be much more rewarding than boring classroom lectures. One of the first things I learned was how to use a quarter to open a locked fire escape window from the outside. While this was technically breaking and entering, we never broke anything. That would have been rude and uncivilized. Instead, we (four friends and I) chose to be creative in our nocturnal wanderings.

For example, we might go into a classroom and viciously rearrange the furniture. We'd put the teacher's desk in the back of the room and turn around all of the chairs. The idea was to see how long such an arrangement would last. It never got past lunch.

Of course, there was the time we inflated a wading pool and filled it full of water in a classroom. That was interesting. It turned out that it was very difficult to empty. They couldn't drag it out because of the narrow doorway and there was no head for a siphon. So, they had to use buckets. That took most of the day. Those small pools sure hold a lot of water! Only one person fell in. At least, they said he fell.

The poor vice principal — the one who handed out detention — was always a prime target. One day, he arrived to find a toilet (old, but mostly clean) in place of his chair. He was not pleased. His office was at the end of a narrow, 10-foot corridor. The interesting architectural features of the corridor were that all the doors: 1) had no windows and 2) all opened into that very confined corridor. Sliding a surplus weather balloon under the door and inflating it only took minutes. You wouldn't think a little air pressure could exert so much force.

Well, let's see, one PSI over a three by six foot door yields over 2,500 pounds of pressure. (My physics teacher would have been proud.) They managed to slide a knife under the door to slice the balloon. A usually reliable source informed me that the deflation resulted in a 10-minute fart.

Surprisingly, the activity that brought the most heat was, to us, the least significant of our actions. It was an underground newspaper. It was just a page or two that was passed around from hand to hand. It was only later that I realized that a prank is just a prank, but an underground newspaper is a threat to authority.

My out-of-school science training was greatly enhanced with various educational items. The chemistry set I got for my birthday was always fun. Of course, everyone knows that the first experiment a young man tries with his new chemistry set is making gunpowder. I was no exception.

Unfortunately, my set was rather limited, so it took me a few years to graduate to contact explosives, napalm, and all the neat things you can do with household chemicals (more explosives, poison gas, etc.). These are clearly important skills that have everyday utility. I would also like to point out that, regardless of reports to the contrary, I only had *one* unintentional explosion in the basement. All the others were definitely done on purpose.

The basement was the proving ground for many ideas. Did you know that, if you throw a dart hard enough at a concrete wall, it will stick? Well, sometimes. The point doesn't even have to be very sharp. Of course, after a few tries, that was pretty obvious.

Shooting a BB gun in the basement was always iffy. The BB would take wild bounces and it never seemed to fail to knock out the overhead fluorescent light. Pellet guns were much better. Pellets would just stick to the concrete wall, but, afterward, you had to scrape them off. Of course, then you had some lead that you could melt with the alcohol burner from the chemistry set to cast fake coins. Even as a youth, I was environmentally responsible and recycled.

Knife throwing was also a skill best learned through practice. The basement was the best place for that because, if the knife completely missed the target, it wasn't likely to ruin the wall — unlike upstairs. I didn't even have to invent creative explanations for unusual circumstances.

Some of my early learning was in the realm of engineering. The automobile coil spark generator is a case in point. It was easy to electrify doorknobs. Too easy. I needed more of a challenge. Running high voltage to different metal parts of the house taught me all about insulation breakdown and other fire hazards. Drilling small holes and sliding wires through walls is just something you can't learn from a textbook.

Reading *SERVO* really made me pause and reflect on one of my first

inventions: The Wheel Mobile. Now, you have to remember that there were no personal computers or robots in those days (except in the movies — Klatu, Barata Niktoe!). So, we had to make do with other everyday items. I rescued an old wheelchair and a lawn mower that someone was discarding. These were good things to take apart and experiment with. That's when I had my epiphany: motorize the wheelchair with the lawn mower engine — such a fabulous idea. Why hadn't anyone thought of it before? I mean, just imagine it.

Handicapped people wouldn't be limited to slow, manual efforts. They'd just zip right along. Why stop there? Wheel Mobiles could be used by anyone. With a five horsepower engine, it would be faster than most motor bikes. You could get right on the expressway! Fast, simple, and inexpensive. What more could you want? You could always use a clear plastic cover of some sort for better aerodynamics and protection against the rain and large insects. I even had a snappy slogan: "See the USA, Wheel Mobile today."

I set out to construct the transportation system of the future. I mounted the engine under the seat. It just fit. I figured that all the air rushing past would cool the muffler enough so that the fabric wouldn't catch on fire. I attached a belt pulley to the shaft and another to the wheel axle. The belt allowed me to connect the vertical engine shaft with the horizontal axle.

Because of the huge wheels, I calculated that the top speed would be about 40 mph. Not shabby at all. I moved the throttle control to the right armrest. The wheelchair already had brakes on each side, next to the big wheels. By selectively pulling on one or the other, I could steer — like a tank stopping one tread to turn. The last thing I did was add a seat belt (actually, just some rope). Hey, 40 mph is fast. I wanted to be safe.

It's with a crystalline memory that I now relate these events. It was evening and my parents were out. (You didn't really think they were aware of my secret invention, did you?) I strapped myself into the Wheel Mobile, reached under the seat and pulled the starter cord. The centrifugal clutch worked perfectly. I was idling and stationary. I pulled back on the throttle a little and the clutch engaged.

Unfortunately, at this point, there were some unforeseen engineering shortcomings in the design that were about to make themselves evident. The clutch engaged too quickly, jerking the Wheel Mobile forward. My hand was on the throttle and the inertia of my arm pulled the throttle back, which increased the throttle, which further increased the inertia on my arm, and so on, and so forth. This was my first experience with an uncontrolled positive feedback servo system. Within a moment, I suspected that it would be my last.

I was propelled forward and promptly smashed into the ping pong table. This knocked me over backward. Of course, this made no difference to the wheelchair because the wheels were so big that they were still in contact with the floor. Speeding under the table, I tried to push the throttle forward only to find that the throttle didn't work.

This was another minor engineering oversight. I used a bicycle brake-cable to control the throttle. As I fell over, my hand moved the whole control by about a half inch. This disengaged the wire from the control. The control was now useless and the cable friction was holding the throttle wide open. It was too late to chastise myself for trying to tighten a Philips head screw with a flat blade screwdriver.

I grabbed at the brakes. I could only find one. I yanked. That wheel stopped, but the other didn't — exactly as designed. Unfortunately, with the throttle malfunction, I was now rapidly spinning in circles under the ping pong table.

I feel that it is important to mention that, at times of near death, the mental and sensory processes are greatly enhanced. I remember seeing writing on the bottom of the table. I couldn't read it, of course, but I do recall thinking, "I don't remember that being there." I also felt that the astronaut centrifuge training must feel similar to my present situation.

After what seemed like forever, I realized that I had to do something or my brain would ooze out of my ears and that would be a real mess to clean up. At that moment, I wanted more than anything to go straight. So, I let go of the brake.

I streaked out from under the table heading (perhaps I should say "bottoming") toward the stairs. Now, I'll stop, I thought confidently. I was, again, in error. Those large wheels ran up the stairs with no effort at all. I went careening down the hall and into the living room, past my surprised sister and her more surprised boyfriend and straight into the far wall.

Please note the geometry of the situation. The Wheel Mobile was on its back, so its base was against the wall. With the forward motion and inertia, the machine again worked as designed and tried to climb the wall. It was only partly successful in this effort. I believe, but am not certain, that it only went up about three feet. I say this because I have the indistinct memory of landing on my head. At this point, the machine stopped because gasoline engines in general and this one, specifically, stall when upside down.

After taking a few moments to return to a cognitive state, I managed to disengage myself from the wreckage. With all deliberate haste I — and my broken dream — withdrew to the basement.

There it sits to this very day. Waiting. Waiting for repairs. Waiting for improvements. Waiting for another chance at destiny. Waiting for the skid marks to fade from the walls. Waiting patiently for hell to freeze over.

Now, a serious thought: If a high schooler did these things today, what would be the result? Pranks in school, playing with knives, air guns, and explosives — outlets for letting off steam are necessary. Are there enough today? If not, the steam can explode.

Perhaps robotics is the answer, with the mental involvement and the option of sheering steel in the combat arena. There are worse things young people can do with their time. SV

Advertiser Index